
LIBERO: 面向终身机器人学习的知识迁移基准

LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning

[†]Bo Liu*, [†]Yifeng Zhu*, [‡]Chongkai Gao*, [†]Yihao Feng

[†]Qiang Liu, [†]Yuke Zhu, ^{†,§}Peter Stone

[†]The University of Texas at Austin, [§]Sony AI, [‡]Tsinghua University

{bliu,yifengz,lqiang,yukez,pstone}@cs.utexas.edu

yihao.ac@gmail.com, gck20@mails.tsinghua.edu.cn

摘要

终身学习为构建通用智能体提供了一条颇具前景的路径：智能体在整个生命周期中持续学习、不断适应。图像与文本领域的传统终身学习问题，主要涉及实体和概念等陈述性知识的迁移；决策场景的终身学习（LLDM）则不同，还需要迁移动作与行为这类程序性知识。为推进 LLDM 研究，我们提出 LIBERO——一个面向机器人操作的终身学习新基准。LIBERO 聚焦 LLDM 的五个核心研究议题：1) 如何高效迁移陈述性知识、程序性知识或二者的混合；2) 如何为 LLDM 设计有效的策略架构与 3) 有效的算法；4) 终身学习器对任务顺序的鲁棒性；5) 模型预训练对 LLDM 的影响。我们开发了一条可扩展的程序化生成流水线，原则上能生成无穷多的任务；为便于基准评测，我们用它构建了四个任务套件（共 130 个任务），用于研究上述议题。所有任务都配有高质量的人类遥操作演示数据，以支持样本高效的学习。大量实验带来了若干有启发、甚至出人意料的发现：顺序微调的前向迁移表现优于现有终身学习方法；没有一种视觉编码器架构能在所有类型的知识迁移上都占优；朴素的监督预训练反而可能拖累智能体在后续 LLDM 中的表现。²

1 引言

打造能胜任广泛任务的通用智能体，是机器学习的长期目标。多任务学习 [10] 是一种途径，但它计算开销大，也难以适应持续的变化。终身学习 [65] 则把学习过程摊到智能体的整个生命周期，更为可行：既利用已有知识加速新任务的学习（前向迁移），也用新学到的知识回头改进旧任务的表现（后向迁移）。

本文为 AI 辅助翻译版本，仅供学习交流；引用请以英文原文为准：arXiv:2306.03310。

*Equal contribution.

²代码与数据集见项目网站 <https://libero-project.github.io>。

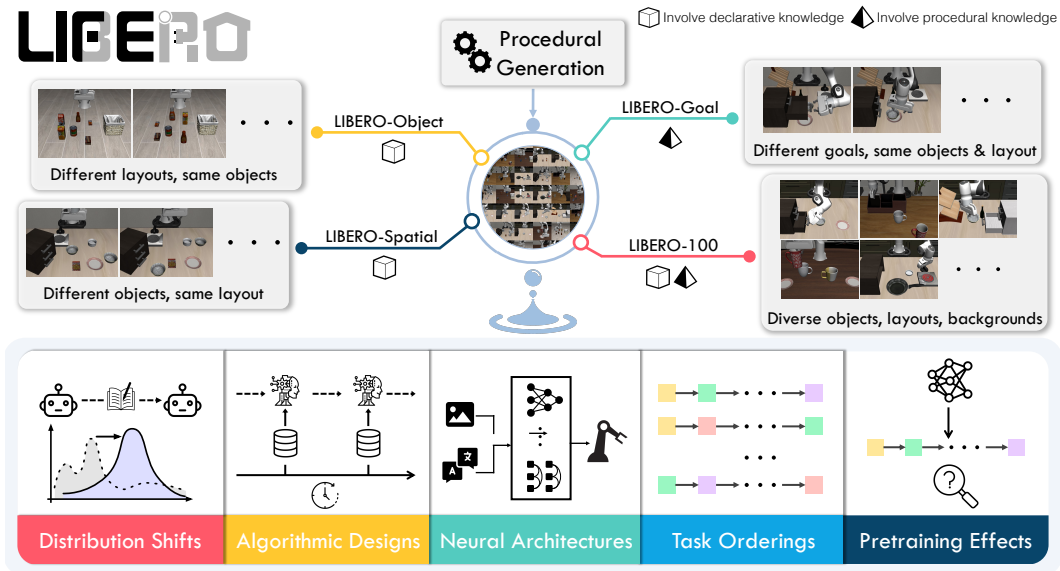


图 1: 上: LIBERO 包含四个程序化生成的任务套件: LIBERO-SPATIAL、LIBERO-OBJECT 与 LIBERO-GOAL 各含 10 个任务, 分别考察空间关系、物体、任务目标三类知识的迁移; LIBERO-100 含 100 个任务, 考察多类知识相互纠缠时的迁移。下: 我们在 LIBERO 上研究 LLDM 的五个核心议题。

终身学习文献的主体集中在视觉或语言任务上, 研究智能体如何迁移关于实体和概念的陈述性知识 [7, 40]; 而决策任务中的知识迁移研究得还很少——那里陈述性知识与程序性知识 (关于如何做事的知识) 混在一起。设想一个场景: 机器人先学会了从冰箱取出果汁, 学过新任务之后却失手了。原因可能是忘了果汁或冰箱的位置 (陈述性知识), 也可能是忘了怎么开冰箱门、怎么抓稳果汁 (程序性知识)。对这种复杂的知识迁移, 我们至今缺少系统、定量的分析手段。

为填补这一空白, 本文提出仿真基准 LIBERO (Lifelong learning BENCHMARK on RObot manipulation tasks), 支持对决策场景的终身学习 (LLDM) 开展系统研究。理想的 LLDM 试验台, 应让智能体在一组不断扩充、彼此共享概念与动作的多样任务上持续学习。LIBERO 靠一条程序化生成流水线满足这一要求: 任务取自机器人操作场景, 彼此共享视觉概念 (陈述性知识) 与交互行为 (程序性知识), 数量可以无限扩充。

为便于基准评测, LIBERO 从人类日常活动 [22] 取材, 生成 130 个语言条件机器人操作任务, 编成四个套件, 分别考察四种分布偏移: 物体类型、物体空间摆放、任务目标, 以及三者的混合 (图 1 上排)。LIBERO 可伸缩、可扩展, 为机器人操作中的终身学习研究量身设计。所有 130 个任务均配有高质量的人类遥操作演示数据, 以支持高效学习。

我们在 LIBERO 上开展了一项初步研究, 考察 LLDM 的五大议题 (图 1 下排): 1) 不同类型分布偏移下的知识迁移; 2) 神经架构设计; 3) 终身学习算法设计; 4) 学习器对任务顺序的鲁棒性; 5) 如何在 LLDM 中利用预训练模型。实验覆盖多种策略架构与多种终身学习算法, 从中得到若干有启发、甚至出人意料观察:

1. 策略架构设计与终身学习算法同等关键。Transformer 架构比循环神经网络更擅长抽象时序信息; 视觉信息丰富 (如物体种类多) 的任务适合 Vision Transformer, 以程序性知识为主的任务则更适合卷积网络。

2. 我们评测的终身学习算法虽能有效防止遗忘，但论前向迁移，普遍不如顺序微调。
3. 实验显示：用预训练语言模型嵌入语义丰富的任务描述来充当任务标识，效果不比直接用任务 ID 好。
4. 在大规模离线数据集上做朴素的监督预训练，反而可能对学习器在 LLDLM 下游的表现产生负面影响。

2 背景

本节介绍问题表述并说明论文关键术语。

2.1 机器人学习的马尔可夫决策过程

机器人学习问题可表述为有限时间范围的马尔可夫决策过程： $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, H, \mu_0, R)$ 。其中 $\mathcal{S}$ 和 $\mathcal{A}$ 分别为机器人的状态空间和动作空间。 μ_0 为初始状态分布， $R: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ 为奖励函数， $\mathcal{T}: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ 为转移函数。本工作采用稀疏奖励设置，以目标谓词 $g: \mathcal{S} \rightarrow \{0, 1\}$ 替换 R 。机器人的目标是学习策略 π 以最大化期望回报： $\max_{\pi} J(\pi) = \mathbb{E}_{s_t, a_t \sim \pi, \mu_0} [\sum_{t=1}^H g(s_t)]$ 。

2.2 机器人终身学习问题

在机器人终身学习问题中，机器人用单一策略 π 依次学习 K 个任务 $\{T^1, \dots, T^K\}$ 。假设 π 以任务为条件，即 $\pi(\cdot | s; T)$ 。每个任务 $T^k \equiv (\mu_0^k, g^k)$ 由初始状态分布 μ_0^k 和目标谓词 g^k 定义。³假设所有任务的 $\mathcal{S}, \mathcal{A}, \mathcal{T}, H$ 相同。至第 k 个任务 T^k ，机器人目标是优化

$$\max_{\pi} J_{\text{LRL}}(\pi) = \frac{1}{k} \sum_{p=1}^k \left[\mathbb{E}_{s_t^p, a_t^p \sim \pi(\cdot; T^p), \mu_0^p} \left[\sum_{t=1}^L g^p(s_t^p) \right] \right]. \quad (1)$$

终身学习的一个关键特征是，智能体在学习任务 T^k 时无法访问之前的 $k-1$ 个任务。

终身模仿学习 由于稀疏奖励强化学习的挑战，我们考虑实际可行的替代设置：用户为序列中的每个任务提供小规模演示数据集。记 $D^k = \{\tau_i^k\}_{i=1}^N$ 为任务 T^k 的 N 条演示。每条 $\tau_i^k = (o_0, a_0, o_1, a_1, \dots, o_{l^k})$ ，其中 $l^k \leq H$ 。 o_t 为机器人的感觉输入，包括感知观测及机器人关节和夹爪的状态信息。实践中，观测 o_t 往往不满足马尔可夫性。沿用部分可观测 MDP 的做法 [25]，我们用聚合的观测历史表示 s_t ，即 $s_t \equiv o_{\leq t} \triangleq (o_0, o_1, \dots, o_t)$ 。由此定义终身模仿学习问题，目标与式 (1) 相同。训练时以行为克隆 [4] 优化如下替代目标函数：

$$\min_{\pi} J_{\text{BC}}(\pi) = \frac{1}{k} \sum_{p=1}^k \mathbb{E}_{o_t, a_t \sim D^p} \left[\sum_{t=0}^{l^p} \mathcal{L}(\pi(o_{\leq t}; T^p), a_t^p) \right], \quad (2)$$

其中 $\mathcal{L}$ 为监督学习损失（如负对数似然损失）， π 为高斯混合模型。类似地，假设在学习 T^k 时，数据集 $\{D^p: p < k\}$ 不完全可用。

3 LLDLM 的研究问题

我们阐述 LLDLM 的五大研究问题，它们激发了 LIBERO 的设计和我们的研究。

(T1) 不同类型知识的迁移 要完成诸如将番茄酱放在篮子里的盘子旁这样的任务，机器人必须理解番茄酱的概念、盘子/篮子的位置以及如何放置番茄酱。一般而言，机器人操作任务需要多种类型的知识，失败原因也就难以定位。我们在第 4.2 节介绍四个任务套件：

³论文全篇，上标/下标用于索引任务/时间步。

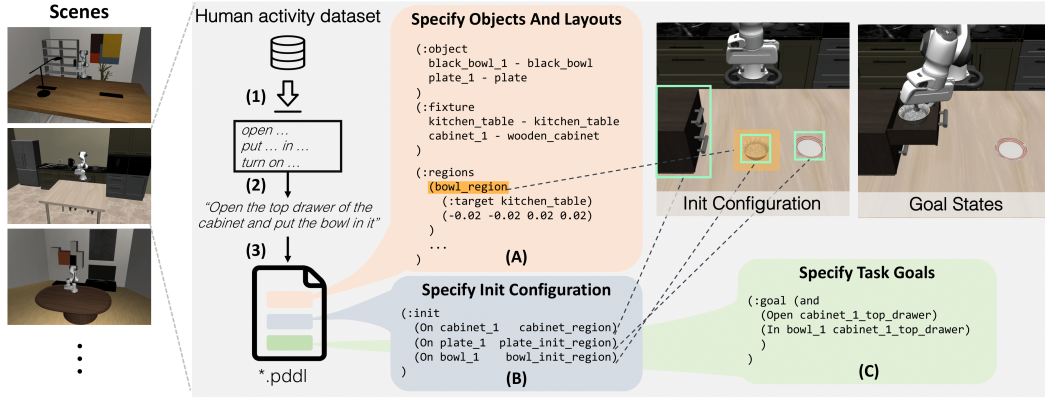


图 2: LIBERO 的程序化生成流程：从大规模人类活动数据集 Ego4D (1) 提取行为模板，用以生成任务指令 (2)；基于任务描述，选择场景并生成 PDDL 描述文件 (3)，指定物体和布局 (A)、物体初始配置 (B) 和任务目标 (C)。

三个套件用于研究空间关系、物体概念和任务目标知识的解耦迁移，一个套件用于研究混合知识类型的迁移。

(T2) 神经网络架构设计 LLDM 的一个重要研究问题是如何设计有效的神经网络架构来抽象多模态观测（图像、语言描述和机器人状态），以及在学习新任务时仅迁移相关知识。

(T3) 终身学习算法设计 给定策略架构，确定为 LLDM 应用什么学习算法至关重要。具体来说，LLDM 按序学习的特性意味着：即使每一步只有轻微遗忘，逐步累积也可能让执行彻底失败。因此，终身学习算法的设计是 LLDM 中的一个开放研究领域。

(T4) 对任务顺序的鲁棒性 任务课程已知会影响策略学习 [6, 48]。然而，现实中的机器人往往无法选择首先遇到哪个任务。因此，良好的终身学习算法应对不同的任务顺序具有鲁棒性。

(T5) 预训练模型的使用 在实践中，机器人部署前很可能已在工厂用大规模数据集预训练过 [28]。然而，预训练是否能以及如何有益于后续的 LLDM 仍不完全明确。

4 LIBERO

本节介绍 LIBERO 的组件：源源不断生成新任务的程序化流程（第 4.1 节）、用于基准测试的四个任务套件（第 4.2 节）、五个算法（第 4.3 节）和三个神经网络架构（第 4.4 节）。

4.1 任务的程序化生成

LLDM 研究需要系统方法创建新任务，维持任务多样性。LIBERO 采用三步程序化生成新任务：1) 从人类活动的语言注解中提取行为模板，基于模板生成自然语言描述的采样任务；2) 基于任务描述指定初始物体分布；3) 用与语言指令一致的命题公式指定任务目标。我们的生成流程基于 Robosuite [76]，这是一个提供无缝集成的模块化操作仿真器。图 2 展示了使用该流程创建任务的示例，下面逐一阐述各个组件。

行为模板与指令生成 人类活动是任务的丰富来源，可激发并生成大量操作任务。我们选择大规模活动数据集 Ego4D [22]，包含具有语言注解的丰富日常活动。我们提取其中

的语言描述并做预处理，总结出一批常用语言模板；再结合仿真器中可用的物体，用模板生成一组自然语言形式的任务描述。例如，从模板「Open ...」可生成指令「Open the drawer of the cabinet」。

初始状态分布 (μ_0) 为指定 μ_0 ，我们首先采样与所提指令中物体/行为相匹配的场景布局。例如，对于指令打开柜子的顶部抽屉并将碗放入，选择厨房场景。随后， μ_0 的细节使用 PDDL 语言生成 [43, 63]。具体地， μ_0 包含物体类别及其放置信息（图 2-(A)）和初始状态（图 2-(B)）。

目标规范 (g) 基于 μ_0 和语言指令，我们使用谓词合取指定任务目标。谓词包括描述物体属性的一元谓词（如 `Open(X)` 或 `TurnOff(X)`）和描述物体间空间关系的二元谓词（如 `On(A, B)` 或 `In(A, B)`）。目标规范的 PDDL 表示示例见图 2-(C)。当所有谓词验证为真时仿真终止。

4.2 任务套件

虽然第 4.1 节的流程支持生成无限数量的任务，我们为基准测试提供固定的任务集合。LIBERO 包含四个任务套件：LIBERO-SPATIAL、LIBERO-OBJECT、LIBERO-GOAL 和 LIBERO-100。前三个套件精心设计以区分陈述性和程序性知识的迁移（如 (T1) 所述），LIBERO-100 则是 100 个任务的套件，涉及相互纠缠的知识迁移。

LIBERO-X LIBERO-SPATIAL、LIBERO-OBJECT 和 LIBERO-GOAL 各包含 10 个任务⁴ 并设计用于研究关于空间信息（陈述性）、物体（陈述性）和任务目标（程序性）的可控知识迁移。具体地，LIBERO-SPATIAL 的所有任务都要求机器人在同一组物体中把碗放到盘子上，但场景里有两个一模一样的碗，区别只在位置或与其他物体的空间关系。因此，要完成 LIBERO-SPATIAL，机器人需要持续学习并记住新的空间关系。LIBERO-OBJECT 中，每个任务要求机器人拾取并放置一种不同的物体。因此，要完成 LIBERO-OBJECT，机器人需要持续学习并记住新的物体类型。LIBERO-GOAL 中的所有任务共享相同的物体和固定的空间关系，但任务目标不同。因此，要完成 LIBERO-GOAL，机器人需要持续学习关于动作和行为的新知识。更多细节见附录 C。

LIBERO-100 LIBERO-100 包含 100 个任务，涉及多样的物体交互和多样的运动技能。本文将 LIBERO-100 分为 90 个短时程任务（LIBERO-90）和 10 个长时程任务（LIBERO-LONG）。LIBERO-90 作为预训练的数据来源 (T5)，LIBERO-LONG 用于终身学习算法的下游评估。

4.3 终身学习算法

我们实现三个代表性的终身学习算法以促进 LLDM 的算法设计研究。具体地，我们实现经验回放 (ER) [13]、弹性权重巩固 (EWC) [33] 和 PACKNET [41]。选择这些方法是因为它们分别对应终身学习的基于记忆、基于正则化和基于动态架构的方法。此外，先前研究 [69] 发现它们是最先进的方法。除这三个方法外，我们还实现了顺序微调 (SEQL) 和多任务学习 (MTL)，分别作为终身学习算法的下界和上界。更多算法细节见附录 B.1。

4.4 神经网络架构

我们实现三个视觉-语言策略网络——RESNET-RNN、RESNET-T 和 ViT-T，集成视觉、时序与语言信息以支持 LLDM。任务的语言指令用预训练 BERT 嵌入 [19] 来编码。RESNET-

⁴10 个任务的套件足以观测到灾难性遗忘，同时保持计算效率。

RNN [42] 使用 ResNet 作为视觉骨干编码逐步视觉观测，使用 LSTM 作为时序骨干处理编码视觉信息序列。语言指令一方面经 FiLM 方法 [50] 融入 ResNet 特征，一方面并入 LSTM 的输入。RESNET-T 架构 [75] 采用类似的基于 ResNet 的视觉骨干，但使用 Transformer 解码器 [66] 作为时序骨干处理 ResNet 输出（视觉令牌的时序序列）。语言嵌入作为独立令牌与视觉令牌一起输入 Transformer。ViT-T 架构 [31]（广泛用于视觉-语言任务），使用 Vision Transformer (ViT) 作为视觉骨干，Transformer 解码器作为时序骨干。语言嵌入在 ViT 和 Transformer 解码器的输入中分别作为独立令牌处理。所有时序骨干为每个决策步骤输出潜向量。我们使用基于高斯混合模型 (GMM) 的输出头计算操作的多模态分布 [8, 42, 68]。最终，机器人通过从输出分布采样末端执行器动作的连续值来执行策略。图 6 展示了三个架构。

对于所有终身学习算法和神经网络架构，我们使用行为克隆 (BC) [4] 训练单个任务的策略（见式 (2)）。BC 的策略学习足够高效，我们因此能在有限计算资源下研究终身学习算法。训练 BC 时，为生成任务套件中的每个任务提供 50 条高质量演示轨迹。演示由人类专家通过 3Dconnexion Spacemouse 遥操作采集。

5 实验

针对第 3 节提出的五个研究课题，我们做了初步实证研究。下文先介绍评估指标，再分析 LIBERO 的实证结果，详细实验配置见附录 D。本研究围绕以下研究问题展开：

- Q1:** 不同架构/终身学习 (LL) 算法在特定分布偏移下表现如何？
- Q2:** 神经网络架构对 LLDM 中知识迁移的影响程度如何，不同架构是否具有可辨识的特殊能力？
- Q3:** 监督终身学习中的现有算法在 LLDM 任务上表现如何？
- Q4:** 语言嵌入对 LLDM 中知识迁移的影响程度如何？
- Q5:** 不同 LL 算法对 LLDM 任务顺序的鲁棒性如何？
- Q6:** 监督预训练能否提升 LLDM 的下游终身学习性能？

5.1 评估指标

我们使用三个指标：FWT（前向迁移）[20]、NBT（负后向迁移）和 AUC（成功率曲线下面积）。所有指标基于成功率计算，文献表明这比训练损失更可靠 [42]（详见附录 E.2）。NBT 越低性能越好，FWT 越高学习越快，AUC 越高综合表现越优。令 $c_{i,j,e}$ 为智能体的成功率：学过 $i-1$ 个前序任务后，当前任务 i 学习 e 轮 ($e \in \{0, 5, \dots, 50\}$) 时在任务 j 上的表现。令 $c_{i,i}$ 为任务 i 上所有评估轮次的最高成功率（即 $c_{i,i} = \max_e c_{i,i,e}$ ）。找出首次达到任务 i 最佳性能的轮次 e_i^* （即 $e_i^* = \arg \min_e c_{i,i,e} = c_{i,i}$ ），并假定对所有 $e \geq e_i^*$ ，有 $c_{i,i,e} = c_{i,i}$ 。⁵ 对于不同任务 $j \neq i$ ，定义 $c_{i,j} = c_{i,j,e_i^*}$ 。三个指标定义如下：

$$\begin{aligned} \text{FWT} &= \sum_{k \in [K]} \frac{\text{FWT}_k}{K}, \quad \text{FWT}_k = \frac{1}{11} \sum_{e \in \{0 \dots 50\}} c_{k,k,e} \\ \text{NBT} &= \sum_{k \in [K]} \frac{\text{NBT}_k}{K}, \quad \text{NBT}_k = \frac{1}{K-k} \sum_{\tau=k+1}^K (c_{k,k} - c_{\tau,k}) \\ \text{AUC} &= \sum_{k \in [K]} \frac{\text{AUC}_k}{K}, \quad \text{AUC}_k = \frac{1}{K-k+1} (\text{FWT}_k + \sum_{\tau=k+1}^K c_{\tau,k}) \end{aligned} \quad (3)$$

⁵实际上，由于学习的方差性，智能体在任务 i 上的性能可能不单调递增。我们保留各轮次 ($\{e\}$) 中保存的最佳检查点，视同智能体在 e_i^* 后停止学习。

图 3 给出这些指标的可视化。

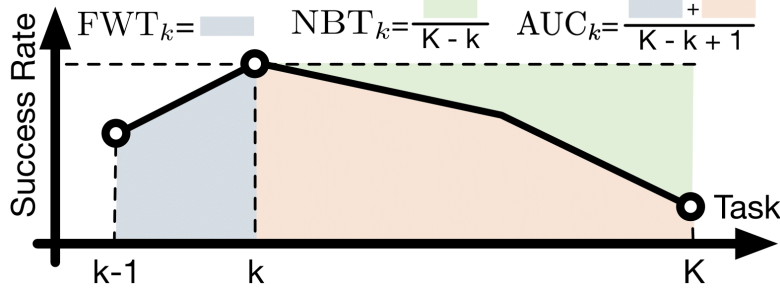


图 3: LLDM 的评估指标。

5.2 实验结果

下文给出实证结果，逐一回答上述研究问题。所有算法、策略架构和任务套件的完整结果详见附录 E.1。

策略神经网络架构研究 (Q1、Q2) 表 1 报告智能体在四个任务套件上使用三种不同神经网络架构的终身学习性能。展示 ER 和 PACKNET 的结果，因为这两个方法在所有任务套件上表现最优。

Policy Arch.	ER			PACKNET		
	FWT($\uparrow$)	NBT($\downarrow$)	AUC($\uparrow$)	FWT($\uparrow$)	NBT($\downarrow$)	AUC($\uparrow$)
LIBERO-LONG (长时程任务)						
RESNET-RNN	0.16 $\pm$ 0.02	0.16 $\pm$ 0.02	0.08 $\pm$ 0.01	0.13 $\pm$ 0.00	0.21 $\pm$ 0.01	0.03 $\pm$ 0.00
RESNET-T	0.48 $\pm$ 0.02	0.32 $\pm$ 0.04	0.32 $\pm$ 0.01	0.22 $\pm$ 0.01	0.08 $\pm$ 0.01	0.25 $\pm$ 0.00
ViT-T	0.38 $\pm$ 0.05	0.29 $\pm$ 0.06	0.25 $\pm$ 0.02	0.36 $\pm$ 0.01	0.14 $\pm$ 0.01	0.34 $\pm$ 0.01
LIBERO-SPATIAL (空间任务)						
RESNET-RNN	0.40 $\pm$ 0.02	0.29 $\pm$ 0.02	0.29 $\pm$ 0.01	0.27 $\pm$ 0.03	0.38 $\pm$ 0.03	0.06 $\pm$ 0.01
RESNET-T	0.65 $\pm$ 0.03	0.27 $\pm$ 0.03	0.56 $\pm$ 0.01	0.55 $\pm$ 0.01	0.07 $\pm$ 0.02	0.63 $\pm$ 0.00
ViT-T	0.63 $\pm$ 0.01	0.29 $\pm$ 0.02	0.50 $\pm$ 0.02	0.57 $\pm$ 0.04	0.15 $\pm$ 0.00	0.59 $\pm$ 0.03
LIBERO-OBJECT (物体任务)						
RESNET-RNN	0.30 $\pm$ 0.01	0.27 $\pm$ 0.05	0.17 $\pm$ 0.05	0.29 $\pm$ 0.02	0.35 $\pm$ 0.02	0.13 $\pm$ 0.01
RESNET-T	0.67 $\pm$ 0.07	0.43 $\pm$ 0.04	0.44 $\pm$ 0.06	0.60 $\pm$ 0.07	0.17 $\pm$ 0.05	0.60 $\pm$ 0.05
ViT-T	0.70 $\pm$ 0.02	0.28 $\pm$ 0.01	0.57 $\pm$ 0.01	0.58 $\pm$ 0.03	0.18 $\pm$ 0.02	0.56 $\pm$ 0.04
LIBERO-GOAL (目标任务)						
RESNET-RNN	0.41 $\pm$ 0.00	0.35 $\pm$ 0.01	0.26 $\pm$ 0.01	0.32 $\pm$ 0.03	0.37 $\pm$ 0.04	0.11 $\pm$ 0.01
RESNET-T	0.64 $\pm$ 0.01	0.34 $\pm$ 0.02	0.49 $\pm$ 0.02	0.63 $\pm$ 0.02	0.06 $\pm$ 0.01	0.75 $\pm$ 0.01
ViT-T	0.57 $\pm$ 0.00	0.40 $\pm$ 0.02	0.38 $\pm$ 0.01	0.69 $\pm$ 0.02	0.08 $\pm$ 0.01	0.76 $\pm$ 0.02

表 1: 使用 ER 和 PACKNET 的三种神经网络架构在四个任务套件上的性能。结果为三组随机种子的平均值，报告均值与标准误。最佳性能用**粗体**标注，如果相对其他架构的改进在统计上显著（双尾 Student t 检验，样本量相等、方差不等， $p = 0.05$ ），则用**紫色**标注。

发现：首先，RESNET-T 和 ViT-T 显著优于 RESNET-RNN，说明 Transformer 在时间维度更优。其次，架构的效果取决于底层算法。用 PACKNET 时，RESNET-T 与 ViT-T 无显著差异（LIBERO-LONG 除外）；用 ER 时，RESNET-T 在大多数任务套件上超过 ViT-T。当网络充分时，ViT 在多物体视觉信息上优于 ResNet（LIBERO-OBJECT 的 MTL 结果为证）。架构设计需兼顾空间与时间信息的处理。

终身学习算法研究（Q1、Q3） 表 2 报告三个终身学习算法的性能，以及 SeqL 和 MTL 基线。所有实验使用 RESNET-T，它是三种策略架构中性能最优的。

Lifelong Algo.	FWT(↑)	NBT(↓)	AUC(↑)	FWT(↑)	NBT(↓)	AUC(↑)
	LIBERO-LONG (长时程)			LIBERO-SPATIAL (空间)		
SeqL	0.54 ± 0.01	0.63 ± 0.01	0.15 ± 0.00	0.72 ± 0.01	0.81 ± 0.01	0.20 ± 0.01
ER	0.48 ± 0.02	0.32 ± 0.04	0.32 ± 0.01	0.65 ± 0.03	0.27 ± 0.03	0.56 ± 0.01
EWC	0.13 ± 0.02	0.22 ± 0.03	0.02 ± 0.00	0.23 ± 0.01	0.33 ± 0.01	0.06 ± 0.01
PACKNET	0.22 ± 0.01	0.08 ± 0.01	0.25 ± 0.00	0.55 ± 0.01	0.07 ± 0.02	0.63 ± 0.00
MTL			0.48 ± 0.01			0.83 ± 0.00
	LIBERO-OBJECT (物体)			LIBERO-GOAL (目标)		
SeqL	0.78 ± 0.04	0.76 ± 0.04	0.26 ± 0.02	0.77 ± 0.01	0.82 ± 0.01	0.22 ± 0.00
ER	0.67 ± 0.07	0.43 ± 0.04	0.44 ± 0.06	0.64 ± 0.01	0.34 ± 0.02	0.49 ± 0.02
EWC	0.56 ± 0.03	0.69 ± 0.02	0.16 ± 0.02	0.32 ± 0.02	0.48 ± 0.03	0.06 ± 0.00
PACKNET	0.60 ± 0.07	0.17 ± 0.05	0.60 ± 0.05	0.63 ± 0.02	0.06 ± 0.01	0.75 ± 0.01
MTL			0.54 ± 0.02			0.80 ± 0.01

表 2: 三个终身学习算法与 SeqL 和 MTL 基线在四个任务套件上的性能对比，策略固定为 RESNET-T。结果为三组随机种子的平均值，报告均值与标准误。最佳性能用**粗体**标注，如果相对其他算法的改进在统计上显著（双尾 Student t 检验，样本量相等、方差不等， $p = 0.05$ ），则用**紫色**标注。

发现：我们观察到一系列对 LLDM 算法设计研究可能有价值的发现：**1)** SeqL 在所有任务套件上的 FWT 都最高。这出人意料——它意味着我们考察的终身学习算法实际上都在损害前向迁移；**2)** PACKNET 在 LIBERO-X 上优于其他终身学习算法，在 LIBERO-LONG 上却被 ER 明显反超，主因是前向迁移偏低。这印证了动态架构方法擅长防遗忘；但 PACKNET 把网络切分成多个子网络，学习单个任务的有效容量随之变小，我们据此推测它的容量不足以学好 LIBERO-LONG；**3)** EWC 不如 SeqL，说明损失项上的正则化反而会拖累智能体在 LLDM 问题上的表现（见附录 E.2）；**4)** 作为重放类方法的 ER 在所有任务套件上都稳健。

语言嵌入作为任务标识的研究（Q4） 为探究语言嵌入在 LLDM 中的作用程度，对比同一终身学习器使用四种不同预训练语言嵌入的性能。选择 BERT [19]、CLIP [52]、GPT-2 [53] 和任务 ID 嵌入。任务 ID 嵌入通过将“Task 5”这类字符串输入预训练 BERT 模型生成。

发现：从表 3 可见，包括任务 ID 嵌入在内，各语言嵌入之间没有统计上的显著差异。我们认为，原因是句子嵌入的作用近似词袋，仅足以区分不同任务。鉴于各嵌入无显著差异，我们默认采用 BERT 嵌入。

Embedding Type	Dimension	FWT($\uparrow$)	NBT($\downarrow$)	AUC($\uparrow$)
BERT	768	0.48 ± 0.02	0.32 ± 0.04	0.32 ± 0.01
CLIP	512	0.52 ± 0.00	0.34 ± 0.01	0.35 ± 0.01
GPT-2	768	0.46 ± 0.01	0.34 ± 0.02	0.30 ± 0.01
Task-ID	768	0.50 ± 0.01	0.37 ± 0.01	0.33 ± 0.01

表 3: 终身学习器在 LIBERO-LONG 上使用四种语言嵌入的性能对比, 其中策略架构固定为 RESNET-T, 终身学习算法固定为 ER。任务 ID 嵌入通过将”Task + ID”输入预训练 BERT 模型获得。结果为三组随机种子的平均值, 报告均值与标准误。最佳性能用**粗体**标注。不同语言嵌入间无统计显著差异。

任务顺序研究 (Q5) 图 4 给出该研究的结果。所有实验使用 RESNET-T 作为神经网络架构, 评估 ER 和 PACKNET 两种算法。如图所示, 两个算法的性能在不同任务顺序下存在差异。这提示未来的重要研究方向: 开发对任务顺序变化具有鲁棒性的算法或架构。

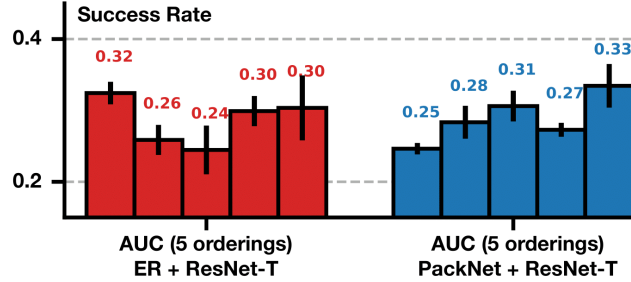


图 4: ER 和 PACKNET 使用 RESNET-T 在五种不同任务顺序下的性能对比。误差条表示固定顺序下性能的标准差。

发现: 从图 4 可见, 不同任务顺序确实导致同一算法的性能差异。具体而言, PACKNET 的这种差异在统计上显著。

预训练对下游 LLDLM 影响的研究 (Q6) 图 5 展示五个算法-架构组合在 LIBERO-LONG 上有/无预训练的性能。预训练用行为克隆: 三种架构各在 LIBERO-100 的 90 个短时程任务上训练 50 轮, 每 5 轮保存检查点, 选最优点作下游初值。

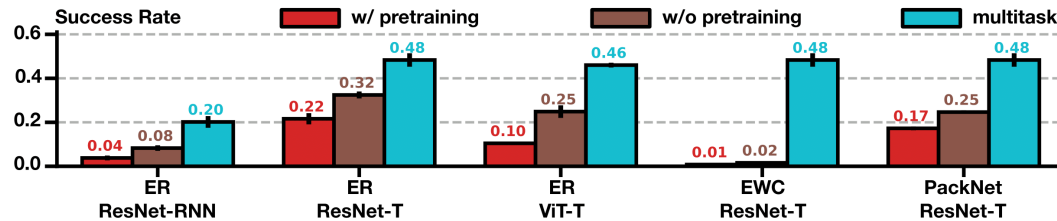


图 5: 不同算法与架构组合在有/无预训练下的性能对比。多任务学习性能作为参考也予以展示。

发现: 我们观察到, 朴素的监督预训练会损害模型在下游终身学习中的表现。结合表 2 的结果 (如朴素顺序微调的前向迁移优于终身学习算法), 这表明需要更优的预训练方法。

注意力可视化： 为更好地理解智能体在终身学习过程中遗忘的知识类型，我们可视化了智能体对各观测图像输入的注意力图。可视化的显著性图和相关讨论见附录 E.4。

6 相关工作

本节概述了终身学习和机器人学习的现有基准。读者可参阅附录 B.1 了解终身学习算法的详细综述。

终身学习基准 先前工作已将标准视觉或语言数据集用于研究终身学习 (LL)。这类工作包括图像分类数据集 (MNIST [18]、CIFAR [34]、ImageNet [17])、分割数据集 (Core50 [38])、自然语言理解数据集 (GLUE [67]、SuperGLUE [59])。除此之外，强化学习 (RL) 中的游戏基准 (如 Atari [46]、XLand [64]、VisDoom [30]) 也被用于 LL 研究。然而，标准监督学习的 LL 不涉及程序性知识转移，游戏中的 RL 问题也不代表人类活动。ContinualWorld [69] 为 LL 修改了 MetaWorld 中的 50 个操作任务。CORA [51] 基于 Atari、Progen [15]、MiniHack [58]、ALFRED [62] 构建了四个终身 RL 基准。F-SIOL-310 [3] 和 OpenLORIS [61] 来自机器人视觉系统的捕获，是具有挑战性的真实世界终身目标学习数据集。先前工作分析了 LL 智能体的不同组件 [45, 70, 21]，但缺乏对机器人操作问题的关注。

机器人学习基准 为应对元学习 (MetaWorld [73])、因果学习 (CausalWorld [1])、多任务学习 [27, 35]、策略泛化至未见对象 [47, 24]、组合学习 [44] 等挑战，已提出多种机器人学习基准。相比终身学习和机器人学习的现有基准，LIBERO 的任务套件精选以解决 LLDLM 的研究课题。该基准包含大量源自日常人类活动的任务，涉及与多样化对象的丰富交互。此外，LIBERO 的任务采用程序化生成，实现基准的可扩展与适应性。LIBERO 提供的高质量人类演示数据集支持高效学习。

7 结论与局限

本文提出 LIBERO——机器人操作领域面向 LLDLM 研究的新基准。LIBERO 内置一条程序化生成流水线，可在仿真器中创建无穷多的操作任务。我们用它构建了 130 个标准化任务，并围绕策略与算法设计开展了系统实验。实验结果指向几个值得继续探索的方向：1) 设计更擅长处理空间信息或时序信息的神经架构；2) 设计前向迁移能力更强的算法；3) 用好预训练，提升终身学习性能。短期内，我们预计 LIBERO 不会带来负面的社会影响；但终身学习器主要向人学习，长远看，如何在 LLDLM 中保护用户隐私 [36] 是必须回答的问题。

参考文献

- [1] Ossama Ahmed, Frederik Träuble, Anirudh Goyal, Alexander Neitz, Yoshua Bengio, Bernhard Schölkopf, Manuel Wüthrich, and Stefan Bauer. Causalworld: A robotic manipulation benchmark for causal structure and transfer learning. *arXiv preprint arXiv:2010.04296*, 2020.
- [2] Ali Ayub and Carter Fendley. Few-shot continual active learning by a robot. *arXiv preprint arXiv:2210.04137*, 2022.
- [3] Ali Ayub and Alan R Wagner. F-siol-310: A robotic dataset and benchmark for few-shot incremental object learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13496–13502. IEEE, 2021.

- [4] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pages 103–129, 1995.
- [5] Eseoghene Ben-Iwhiwhu, Saptarshi Nath, Praveen K Pilly, Soheil Kolouri, and Andrea Soltoggio. Lifelong reinforcement learning with modulating masks. *arXiv preprint arXiv:2212.11110*, 2022.
- [6] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.
- [7] Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-Jussa. Continual lifelong learning in natural language processing: A survey. *arXiv preprint arXiv:2012.09823*, 2020.
- [8] Christopher M Bishop. Mixture density networks. 1994.
- [9] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.
- [10] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.
- [11] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.
- [12] Arslan Chaudhry, Marc’Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.
- [13] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc’Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.
- [14] Brian Cheung, Alexander Terekhov, Yubei Chen, Pulkit Agrawal, and Bruno Olshausen. Superposition of many models into one. *Advances in neural information processing systems*, 32, 2019.
- [15] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056. PMLR, 2020.
- [16] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.
- [17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

- [18] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [20] Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don’t forget, there is more than forgetting: new metrics for continual learning. *arXiv preprint arXiv:1810.13166*, 2018.
- [21] Beyza Ermis, Giovanni Zappella, Martin Wistuba, and Cédric Archambeau. Memory efficient continual learning with transformers. 2022.
- [22] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.
- [23] Sam Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. *ArXiv*, abs/1711.00138, 2017.
- [24] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.
- [25] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*, 2015.
- [26] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- [27] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.
- [28] Leslie Pack Kaelbling. The foundation of efficient robot learning. *Science*, 369(6506):915–916, 2020.
- [29] Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16071–16080, 2022.
- [30] Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE conference on computational intelligence and games (CIG)*, pages 1–8. IEEE, 2016.

- [31] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.
- [32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [33] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
- [34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [35] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023.
- [36] B. Liu, Qian Liu, and Peter Stone. Continual learning and private unlearning. In *CoLLAs*, 2022.
- [37] Hao Liu and Huaping Liu. Continual learning with recursive gradient optimization. *arXiv preprint arXiv:2201.12522*, 2022.
- [38] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, pages 17–26. PMLR, 2017.
- [39] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.
- [40] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.
- [41] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.
- [42] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.
- [43] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. Pddl-the planning domain definition language. 1998.

- [44] Jorge A Mendez, Marcel Hussing, Meghna Gummadi, and Eric Eaton. Composuite: A compositional reinforcement learning benchmark. *arXiv preprint arXiv:2207.04136*, 2022.
- [45] Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture matters in continual learning. *arXiv preprint arXiv:2202.00275*, 2022.
- [46] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [47] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021.
- [48] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *arXiv preprint arXiv:2003.04960*, 2020.
- [49] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.
- [50] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [51] Sam Powers, Eliot Xing, Eric Kolve, Roozbeh Mottaghi, and Abhinav Gupta. Cora: Benchmarks, baselines, and metrics as a platform for continual reinforcement learning agents. *arXiv preprint arXiv:2110.10067*, 2021.
- [52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [53] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [54] Amanda Rios and Laurent Itti. Lifelong learning without a task oracle. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 255–263. IEEE, 2020.
- [55] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

- [56] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
- [57] Gobinda Saha, Isha Garg, Aayush Ankit, and Kaushik Roy. Space: Structured compression and sharing of representational space for continual learning. *IEEE Access*, 9:150480–150494, 2021.
- [58] Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich Küttler, Edward Grefenstette, and Tim Rocktäschel. Minihack the planet: A sandbox for open-ended reinforcement learning research. *arXiv preprint arXiv:2109.13202*, 2021.
- [59] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.
- [60] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pages 4528–4537. PMLR, 2018.
- [61] Qi She, Fan Feng, Xinyue Hao, Qihan Yang, Chuanlin Lan, Vincenzo Lomonaco, Xuesong Shi, Zhengwei Wang, Yao Guo, Yimin Zhang, et al. Openloris-object: A robotic vision dataset and benchmark for lifelong deep learning. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 4767–4773. IEEE, 2020.
- [62] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.
- [63] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on Robot Learning*, pages 477–490. PMLR, 2022.
- [64] Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, et al. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*, 2021.
- [65] Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46, 1995.
- [66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

- [67] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.
- [68] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.
- [69] Maciej Wołczyk, Michal Zajkac, Razvan Pascanu, Lukasz Kuciński, and Piotr Miloś. Continual world: A robotic benchmark for continual reinforcement learning. In *Neural Information Processing Systems*, 2021.
- [70] Maciej Wołczyk, Michal Zajkac, Razvan Pascanu, Lukasz Kuciński, and Piotr Miloś. Disentangling transfer in continual reinforcement learning. *ArXiv*, abs/2209.13900, 2022.
- [71] Lemeng Wu, Bo Liu, Peter Stone, and Qiang Liu. Firefly neural architecture descent: a general approach for growing neural networks. *Advances in Neural Information Processing Systems*, 33:22373–22383, 2020.
- [72] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.
- [73] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.
- [74] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9046–9056, 2022.
- [75] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. *arXiv preprint arXiv:2210.11339*, 2022.
- [76] Yuke Zhu, Josiah Wong, Ajay Mandlekar, and Roberto Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

检查清单

1. 所有作者...
 - (a) 摘要和引言中的主要声明是否准确反映了论文的贡献和范围? [Yes]
 - (b) 你是否描述了工作的局限性? [Yes]
 - (c) 你是否讨论了工作可能产生的负面社会影响? [Yes]
 - (d) 你是否阅读了伦理审查指南并确保论文符合其要求? [Yes]
2. 如果包含理论结果...
 - (a) 你是否陈述了所有理论结果的完整假设集合? [N/A]
 - (b) 你是否包含了所有理论结果的完整证明? [N/A]
3. 如果开展了实验（如基准评估） ...
 - (a) 你是否包含了复现主要实验结果所需的代码、数据和说明（在补充材料中或作为 URL)? [Yes]
 - (b) 你是否指定了所有训练详情（如数据划分、超参数及其选择方式)? [Yes]
 - (c) 你是否报告了误差条（如关于多次运行实验后的随机种子)? [Yes]
 - (d) 你是否包含了计算总量和所用资源类型（如 GPU 类型、内部集群或云提供商)? [Yes]
4. 如果使用了现有资产（如代码、数据、模型）或策划/发布新资产...
 - (a) 如果工作使用了现有资产，你是否引用了创建者? [Yes]
 - (b) 你是否提及了资产的许可证? [N/A]
 - (c) 你是否在补充材料中或作为 URL 包含了新资产? [Yes]
 - (d) 你是否讨论了如何从你所使用/策划的数据的人员那里获得同意? [N/A]
 - (e) 你是否讨论了你所使用/策划的数据是否包含个人可识别信息或冒犯性内容? [N/A]
5. 如果使用了众包或开展了涉及人类受试者的研究...
 - (a) 你是否包含了给予参与者的完整指示文本和屏幕截图（如适用)? [N/A]
 - (b) 你是否描述了任何潜在的参与者风险及机构审查委员会（IRB）批准相关链接（如适用)? [N/A]
 - (c) 你是否包含了支付给参与者的估计每小时工资和参与者的补偿总金额? [N/A]

A 实现的神经网络架构与终身学习算法

Neural Policy Arch.	RESNET-RNN
	RESNET-T
	ViT-T
Lifelong Learning Algo.	SEQL
	EWC [33]
	ER [13]
	PACKNET [41]
	MTL

表 4: 在 LIBERO 中实现的神经策略架构与终身学习算法。

A.1 神经网络架构

第 4.4 节阐述了实验中的神经网络架构，包括 RESNET-RNN、RESNET-T 和 ViT-T。图 6 说明各架构设计。表 5、6 和 7 列举各架构的超参数。

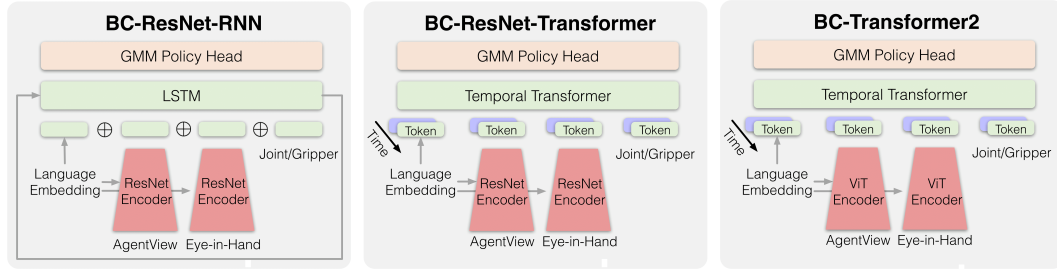


图 6: RESNET-RNN、RESNET-T 与 ViT-T 的架构示意。注意，各架构融合语言嵌入的方式不同。

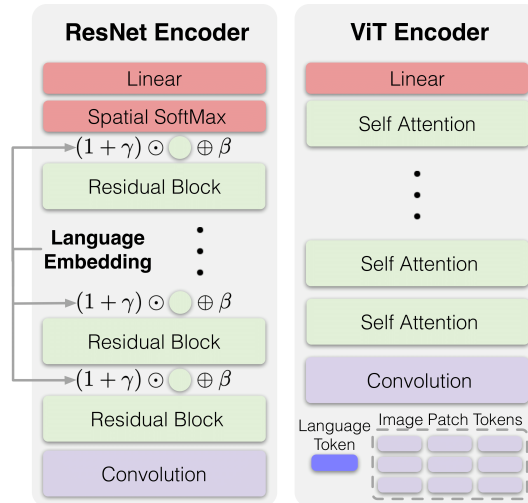


图 7: 图像编码器：基于 ResNet 的编码器与基于视觉 Transformer 的编码器。

B 计算资源

所有实验在单块 Nvidia A100 或 A40 GPU (CUDA 11.7) 加 8-16 个 CPU 核心上完成训练与评估。

Variable	Value
resnet_image_embed_size	64
text_embed_size	32
rnn_hidden_size	1024
rnn_layer_num	2
rnn_dropout	0.0

表 5: RESNET-RNN 的超参数。

Variable	Value
extra_info_hidden_size	128
img_embed_size	64
transformer_num_layers	4
transformer_num_heads	6
transformer_head_output_size	64
transformer_mlp_hidden_size	256
transformer_dropout	0.1
transformer_max_seq_len	10

表 6: RESNET-T 的超参数。

Variable	Value
extra_info_hidden_size	128
img_embed_size	128
spatial_transformer_num_layers	7
spatial_transformer_num_heads	8
spatial_transformer_head_output_size	120
spatial_transformer_mlp_hidden_size	256
spatial_transformer_dropout	0.1
spatial_down_sample_embed_size	64
temporal_transformer_input_size	null
temporal_transformer_num_layers	4
temporal_transformer_num_heads	6
temporal_transformer_head_output_size	64
temporal_transformer_mlp_hidden_size	256
temporal_transformer_dropout	0.1
temporal_transformer_max_seq_len	10

表 7: ViT-T 的超参数。

B.1 终身学习算法

终身学习 (LL) 研究智能体如何在无限的任务序列中持续获取并保留知识, 同时不灾难性地遗忘已有知识。近期文献提出三类主要方法应对深度学习中的灾难性遗忘: 动态架构方法、基于正则化的方法与经验回放方法。虽然也有工作探索几类方法的组合 [2, 29, 54] 或新策略 [74, 57, 14], 我们的基准聚焦这三个基本方向, 剖析它们在机器人学习任务中的优缺点。

动态架构方法逐步扩展学习模型以融入新知识 [56, 72, 41, 26, 71, 5]。基于正则化的方法在学习新任务时, 将学习器正则化到先前的检查点 [33, 11, 60, 37]。经验回放方法保存先前任务的演示数据, 与新数据一起重放以巩固智能体的记忆 [13, 39, 12, 9]。关于 LL 方法的综合评述, 我们推荐读者参考相关综述 [16, 49]。

下面说明我们实现的三个终身学习算法。

ER 经验回放 (ER) [13] 是一种**基于记忆**的方法, 维护一个存放先前任务样本的记忆缓冲区, 供学习新任务时复用。每个任务的策略学完后, ER 把部分数据存入缓冲区。训练新任务时, ER 从记忆中采样数据, 并将其与当前任务的训练数据结合, 使训练数据近似代表全任务数据的经验分布。在实现中, 每个任务训练后, 我们使用回放缓冲区存储部分训练数据 (最多 1000 条轨迹)。新任务训练的每次迭代中, 均匀采样固定数量的回放数据 (32 条轨迹) 与当前任务的训练批次结合。

EWC 弹性权重巩固 (EWC) [33] 是一种**基于正则化**的方法, 通过添加正则化项来约束神经网络更新, 保持原始单任务学习目标。具体地, EWC 使用 Fisher 信息矩阵量化每个神经网络参数的重要性。任务 k 的损失函数为:

$$\mathcal{L}_k^{EWC}(\theta) = \mathcal{L}_K^{BC}(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta_{k-1,i}^*)^2,$$

其中 λ 是惩罚超参数, 系数 F_i 是 Fisher 信息矩阵的对角线元素: $F_k = \mathbb{E}_{s \sim \mathcal{D}_k} \mathbb{E}_{a \sim p_\theta(\cdot|s)} (\nabla_{\theta_k} \log p_{\theta_k}(a|s))^2$ 。本工作采用 EWC 的在线更新版本, 沿终身学习过程用指数移动平均更新 Fisher 信息矩阵, 并使用经验估计稳定矩阵估计。具体地, 实际使用的 Fisher 信息矩阵估计为 $\tilde{F}_k = \gamma F_{k-1} + (1 - \gamma) F_k$, 其中 $F_k = \mathbb{E}_{(s,a) \sim \mathcal{D}_k} (\nabla_{\theta_k} \log p_{\theta_k}(a|s))^2$, k 为任务编号。我们设置 $\gamma = 0.9$ 和 $\lambda = 5 \cdot 10^4$ 。

PackNet PACKNET [41] 是一种**动态架构**方法, 防止对先前任务重要的参数在终身学习中被改动。为此, PACKNET 对网络反复执行训练、剪枝、微调与冻结。该方法理论上完全避免灾难性遗忘, 代价是每学一个新任务, 可用参数就少一些。剪枝分两步: 先正常训练网络, 训练结束时按固定比例 (我们取 25%) 挑出最重要的参数, 其余剪掉; 再微调选中的参数, 微调完冻结。实现上我们遵循原始论文 [41], 不训练任何偏置项与归一化层; 微调轮数与训练相同 (50 轮)。注意, 所有评估指标都在微调阶段之前计算。

C LIBERO 任务套件设计

C.1 任务套件

图 8- 11 分别展示四个任务套件的全部任务。各图均展示任务的目标状态，只有图 8 例外：LIBERO-SPATIAL 各任务的目标相同，故展示初始状态。

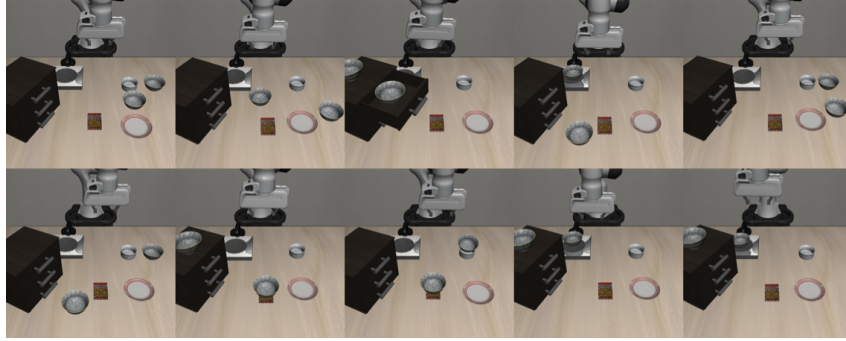


图 8: LIBERO-SPATIAL



图 9: LIBERO-OBJECT

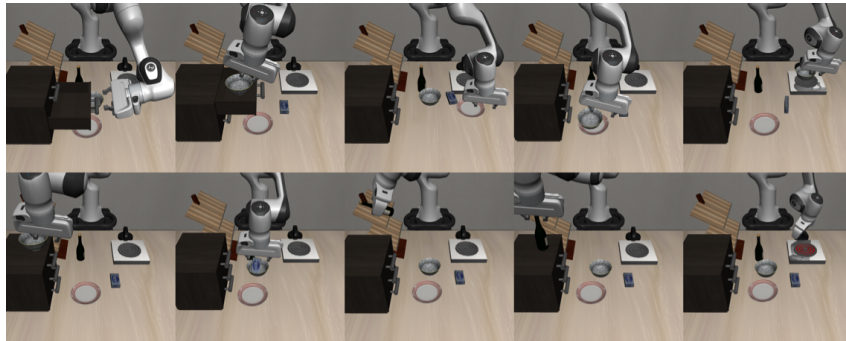


图 10: LIBERO-Goal

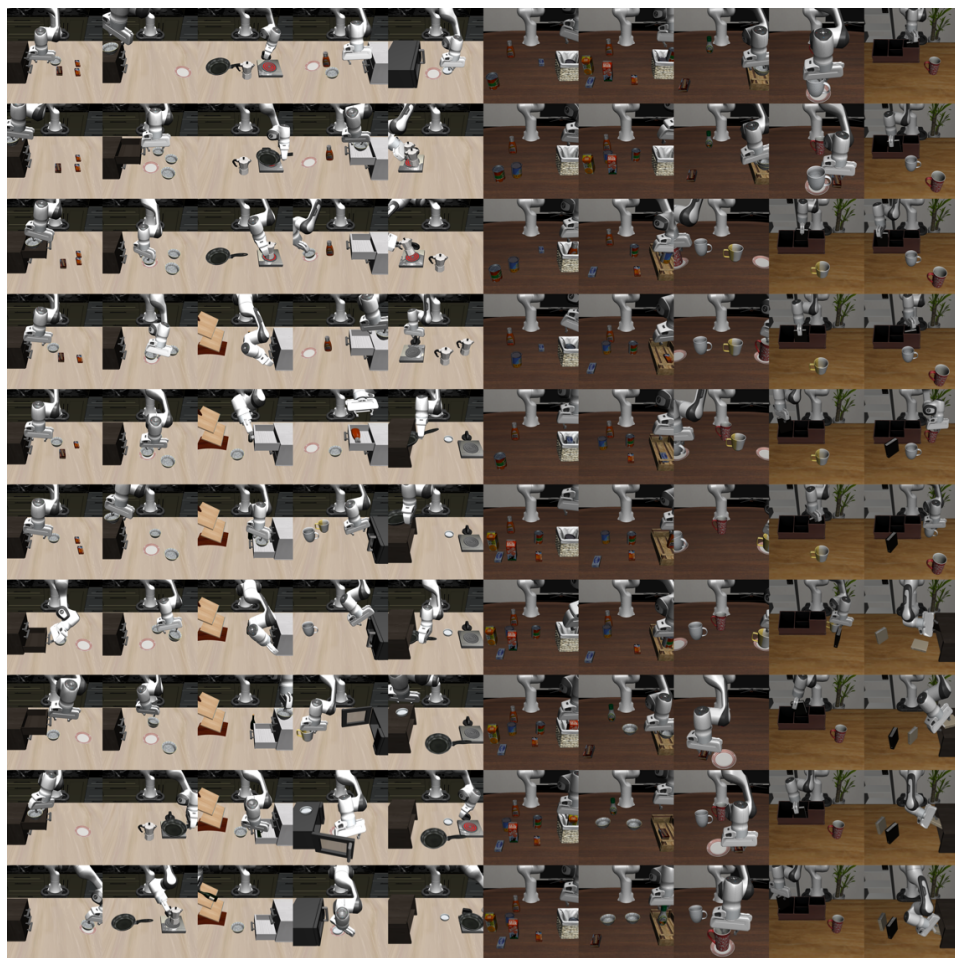


图 11: LIBERO-100

C.2 基于 PDDL 的场景描述文件

下面给出一个基于 PDDL 的场景描述文件的完整示例，对应于图 2 所示的任务。

示例任务： *Open the top drawer of the cabinet and put the bowl in it* (中文：打开柜子最上层抽屉，把碗放进去)。

```
(define (problem LIBERO_Kitchen_Tabletop_Manipulation)
  (:domain robosuite)
  (:language open the top drawer of the cabinet and put the bowl in it)
  (:regions
    (wooden_cabinet_init_region
      (:target kitchen_table)
      (:ranges (
        (-0.01 -0.31 0.01 -0.29)
      ))
      (:yaw_rotation (
        (3.141592653589793 3.141592653589793)
      ))
    )
    (akita_black_bowl_init_region
      (:target kitchen_table)
      (:ranges (
        (-0.025 -0.025 0.025 0.025)
      ))
      (:yaw_rotation (
        (0.0 0.0)
      ))
    )
    (plate_init_region
      (:target kitchen_table)
      (:ranges (
        (-0.025 0.225 0.025 0.275)
      ))
      (:yaw_rotation (
        (0.0 0.0)
      ))
    )
  )
  (top_side
```

```

        (:target wooden_cabinet_1)
      )
      (top_region
        (:target wooden_cabinet_1)
      )
      (middle_region
        (:target wooden_cabinet_1)
      )
      (bottom_region
        (:target wooden_cabinet_1)
      )
    )

  (:fixtures
    kitchen_table - kitchen_table
    wooden_cabinet_1 - wooden_cabinet
  )

  (:objects
    akita_black_bowl_1 - akita_black_bowl
    plate_1 - plate
  )

  (:obj_of_interest
    wooden_cabinet_1
    akita_black_bowl_1
  )

  (:init
    (On akita_black_bowl_1 kitchen_table_akita_black_bowl_init_region)
    (On plate_1 kitchen_table_plate_init_region)
    (On wooden_cabinet_1 kitchen_table_wooden_cabinet_init_region)
  )

  (:goal
    (And (Open wooden_cabinet_1_top_region)
      (In akita_black_bowl_1 wooden_cabinet_1_top_region)
    )
  )
)

```

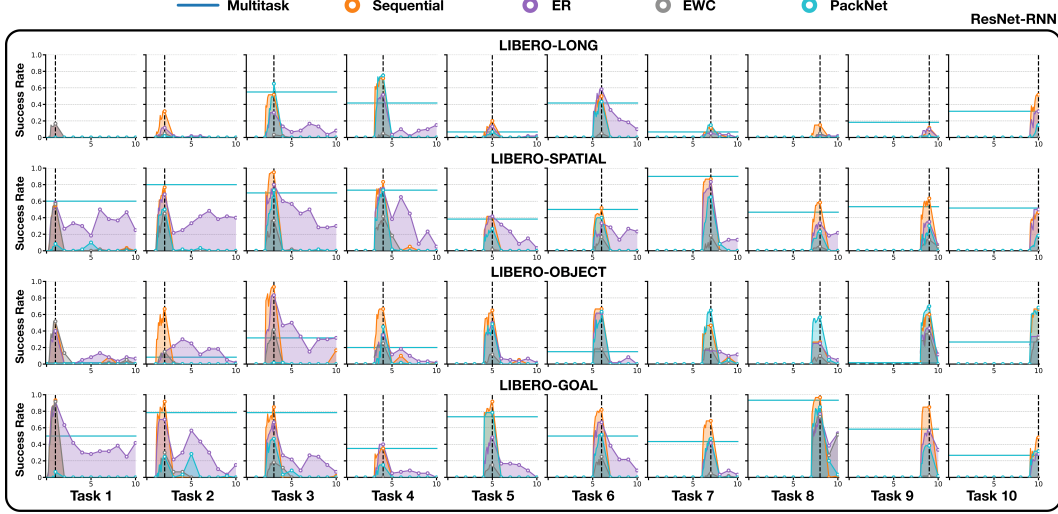


图 12: 我们在图 12 中使用 RESNET-RNN 策略架构对比不同算法的性能。 y 轴表示成功率, x 轴显示智能体在学习过程中对特定任务套件中 10 个任务中每个任务的性能。例如, 图中左上方的图显示智能体顺序学习 10 个任务时对第一个任务的性能。

D 实验设置

我们评估五种终身学习算法: SEQL (顺序学习基线, 直接逐个学习各个任务)、MTL (多任务学习基线, 同时学习所有任务)、正则化方法 EWC [33]、经验回放方法 ER [13] 和动态架构方法 PACKNET [41]。SEQL 和 MTL 分别代表终身学习的下界与上界, 其他三种方法覆盖主要技术路线。我们采用三种视觉-语言策略架构: RESNET-RNN、RESNET-T 和 ViT-T, 三者聚合空间或时序信息的方式不同 (详见附录 A.1)。每个任务训练 50 个 epoch, 每 5 个 epoch 评估一次平均成功率 (20 条最长 600 步的测试轨迹)。采用 Adam 优化器 [32], 批大小 32, 学习率余弦衰减从 0.0001 到 0.00001。按 Robomimic 惯例 [42], 选择成功率最高的检查点作为最终策略。完成 50 个 epoch 训练后, 取最佳检查点的智能体在所有已学任务上评估, 每个任务 20 条测试轨迹。所有策略网络按浮点运算数 (FLOPS) 对齐: 所有架构约 13.5G FLOPS。对于算法、架构和任务套件的每种组合, 运行 3 次终身学习方法, 使用随机种子 100, 200, 300 (共 180 个实验)。具体实现的算法与架构见表 4。

E 额外实验结果

E.1 完整结果

表 8 列出三种终身学习算法 (EWC、ER、PACKNET) 和三种策略架构 (RESNET-RNN、RESNET-T、ViT-T) 在四个任务套件上的全部结果。

为展示每个终身学习智能体在学习过程中的性能演变, 我们绘制两组对比图: 一组固定策略架构、比较不同终身学习算法 (图 12、13、14); 另一组固定终身学习算法、比较不同策略架构 (图 15、16、17)

Algo.	Policy Arch.	FWT($\uparrow$)	NBT($\downarrow$)	AUC($\uparrow$)	FWT($\uparrow$)	NBT($\downarrow$)	AUC($\uparrow$)
		LIBERO-LONG			LIBERO-SPATIAL		
SEQL	RESNET-RNN	0.24 $\pm$ 0.02	0.28 $\pm$ 0.01	0.07 $\pm$ 0.01	0.50 $\pm$ 0.01	0.61 $\pm$ 0.01	0.14 $\pm$ 0.01
	RESNET-T	0.54 $\pm$ 0.01	0.63 $\pm$ 0.01	0.15 $\pm$ 0.00	0.72 $\pm$ 0.01	0.81 $\pm$ 0.01	0.20 $\pm$ 0.01
	ViT-T	0.44 $\pm$ 0.04	0.50 $\pm$ 0.05	0.13 $\pm$ 0.01	0.63 $\pm$ 0.02	0.76 $\pm$ 0.01	0.16 $\pm$ 0.01
ER	RESNET-RNN	0.16 $\pm$ 0.02	0.16 $\pm$ 0.02	0.08 $\pm$ 0.01	0.40 $\pm$ 0.02	0.29 $\pm$ 0.02	0.29 $\pm$ 0.01
	RESNET-T	0.48 $\pm$ 0.02	0.32 $\pm$ 0.04	0.32 $\pm$ 0.01	0.65 $\pm$ 0.03	0.27 $\pm$ 0.03	0.56 $\pm$ 0.01
	ViT-T	0.38 $\pm$ 0.05	0.29 $\pm$ 0.06	0.25 $\pm$ 0.02	0.63 $\pm$ 0.01	0.29 $\pm$ 0.02	0.50 $\pm$ 0.02
EWC	RESNET-RNN	0.02 $\pm$ 0.00	0.04 $\pm$ 0.01	0.00 $\pm$ 0.00	0.14 $\pm$ 0.02	0.23 $\pm$ 0.02	0.03 $\pm$ 0.00
	RESNET-T	0.13 $\pm$ 0.02	0.22 $\pm$ 0.03	0.02 $\pm$ 0.00	0.23 $\pm$ 0.01	0.33 $\pm$ 0.01	0.06 $\pm$ 0.01
	ViT-T	0.05 $\pm$ 0.02	0.09 $\pm$ 0.03	0.01 $\pm$ 0.00	0.32 $\pm$ 0.03	0.48 $\pm$ 0.03	0.06 $\pm$ 0.01
PACKNET	RESNET-RNN	0.13 $\pm$ 0.00	0.21 $\pm$ 0.01	0.03 $\pm$ 0.00	0.27 $\pm$ 0.03	0.38 $\pm$ 0.03	0.06 $\pm$ 0.01
	RESNET-T	0.22 $\pm$ 0.01	0.08 $\pm$ 0.01	0.25 $\pm$ 0.00	0.55 $\pm$ 0.01	0.07 $\pm$ 0.02	0.63 $\pm$ 0.00
	ViT-T	0.36 $\pm$ 0.01	0.14 $\pm$ 0.01	0.34 $\pm$ 0.01	0.57 $\pm$ 0.04	0.15 $\pm$ 0.00	0.59 $\pm$ 0.03
MTL	RESNET-RNN			0.20 $\pm$ 0.01			0.61 $\pm$ 0.00
	RESNET-T			0.48 $\pm$ 0.01			0.83 $\pm$ 0.00
	ViT-T			0.46 $\pm$ 0.00			0.79 $\pm$ 0.01
		LIBERO-OBJECT			LIBERO-GOAL		
SEQL	RESNET-RNN	0.48 $\pm$ 0.03	0.53 $\pm$ 0.04	0.15 $\pm$ 0.01	0.61 $\pm$ 0.01	0.73 $\pm$ 0.01	0.16 $\pm$ 0.00
	RESNET-T	0.78 $\pm$ 0.04	0.76 $\pm$ 0.04	0.26 $\pm$ 0.02	0.77 $\pm$ 0.01	0.82 $\pm$ 0.01	0.22 $\pm$ 0.00
	ViT-T	0.76 $\pm$ 0.03	0.73 $\pm$ 0.03	0.27 $\pm$ 0.02	0.75 $\pm$ 0.01	0.85 $\pm$ 0.01	0.20 $\pm$ 0.01
ER	RESNET-RNN	0.30 $\pm$ 0.01	0.27 $\pm$ 0.05	0.17 $\pm$ 0.05	0.41 $\pm$ 0.00	0.35 $\pm$ 0.01	0.26 $\pm$ 0.01
	RESNET-T	0.67 $\pm$ 0.07	0.43 $\pm$ 0.04	0.44 $\pm$ 0.06	0.64 $\pm$ 0.01	0.34 $\pm$ 0.02	0.49 $\pm$ 0.02
	ViT-T	0.70 $\pm$ 0.02	0.28 $\pm$ 0.01	0.57 $\pm$ 0.01	0.57 $\pm$ 0.00	0.40 $\pm$ 0.02	0.38 $\pm$ 0.01
EWC	RESNET-RNN	0.17 $\pm$ 0.04	0.23 $\pm$ 0.04	0.06 $\pm$ 0.01	0.16 $\pm$ 0.01	0.22 $\pm$ 0.01	0.06 $\pm$ 0.01
	RESNET-T	0.56 $\pm$ 0.03	0.69 $\pm$ 0.02	0.16 $\pm$ 0.02	0.32 $\pm$ 0.02	0.48 $\pm$ 0.03	0.06 $\pm$ 0.00
	ViT-T	0.57 $\pm$ 0.03	0.64 $\pm$ 0.03	0.23 $\pm$ 0.00	0.32 $\pm$ 0.04	0.45 $\pm$ 0.04	0.07 $\pm$ 0.01
PACKNET	RESNET-RNN	0.29 $\pm$ 0.02	0.35 $\pm$ 0.02	0.13 $\pm$ 0.01	0.32 $\pm$ 0.03	0.37 $\pm$ 0.04	0.11 $\pm$ 0.01
	RESNET-T	0.60 $\pm$ 0.07	0.17 $\pm$ 0.05	0.60 $\pm$ 0.05	0.63 $\pm$ 0.02	0.06 $\pm$ 0.01	0.75 $\pm$ 0.01
	ViT-T	0.58 $\pm$ 0.03	0.18 $\pm$ 0.02	0.56 $\pm$ 0.04	0.69 $\pm$ 0.02	0.08 $\pm$ 0.01	0.76 $\pm$ 0.02
MTL	RESNET-RNN			0.10 $\pm$ 0.03			0.59 $\pm$ 0.00
	RESNET-T			0.54 $\pm$ 0.02			0.80 $\pm$ 0.01
	ViT-T			0.78 $\pm$ 0.02			0.82 $\pm$ 0.01

表 8: 表中展示所有网络和算法在四个任务套件上的结果。对每个任务套件, 前三高 AUC 分值用**品红** (最佳)、**浅品红** (第二)、**超浅品红** (第三) 标记。

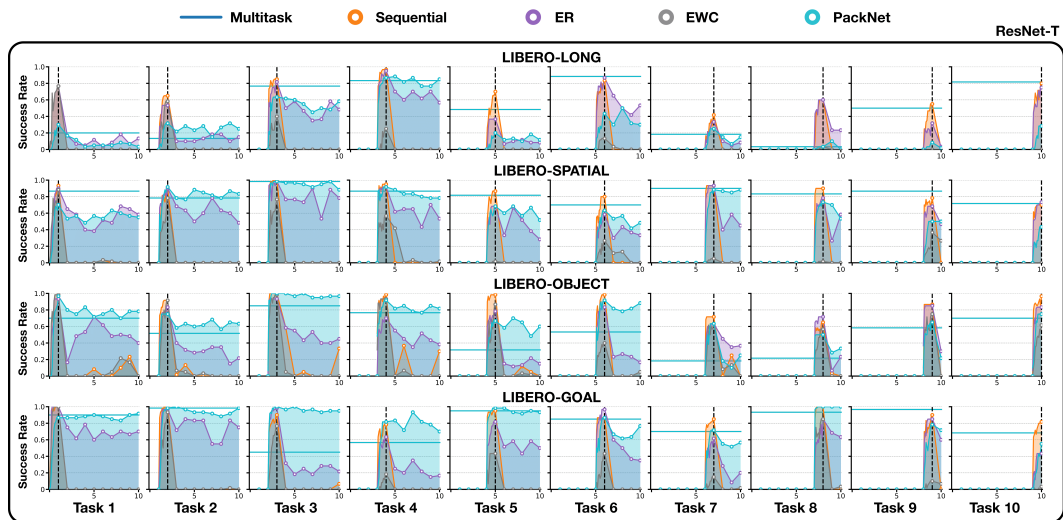


图 13: 使用 RESNET-T 策略架构对比不同算法。 y 轴表示成功率, x 轴显示智能体在学习过程中对给定任务套件中 10 个任务中每个任务的性能。例如, 左上角的图显示智能体顺序学习 10 个任务时对第一个任务的性能。

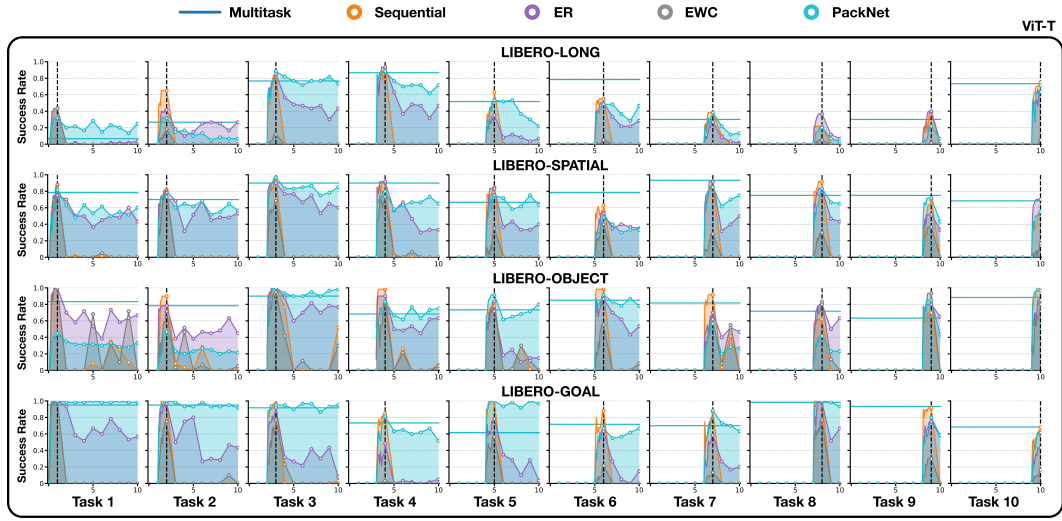


图 14: 使用 ViT-T 策略架构对比不同算法。y 轴表示成功率, x 轴显示智能体在学习过程中对给定任务套件中 10 个任务的性能。例如, 左上角的图说明智能体顺序学习 10 个任务时对第一个任务的性能。

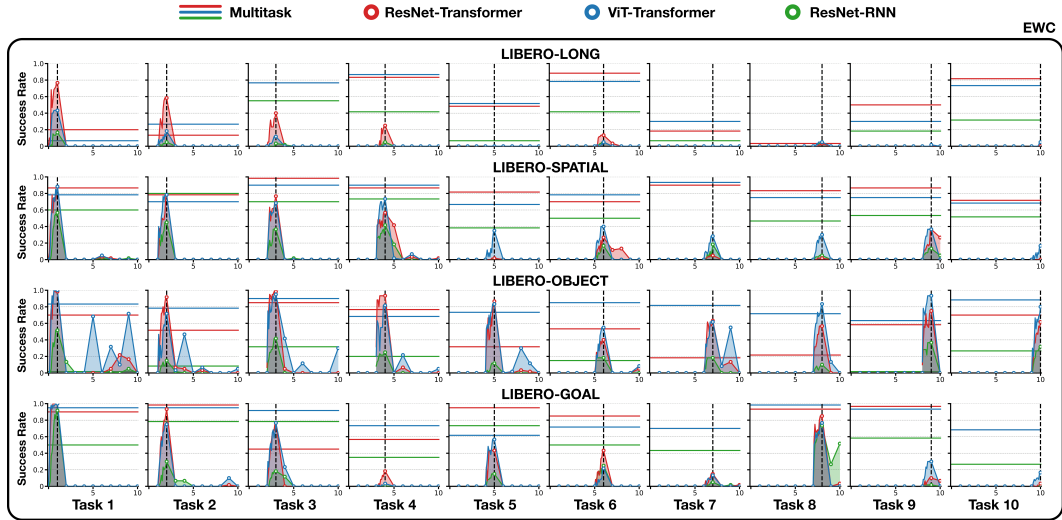


图 15: 使用 EWC 算法对比不同架构。y 轴为成功率, x 轴显示智能体在学习过程中对给定任务套件中 10 个任务的性能。例如, 左上方的图显示智能体顺序学习 10 个任务时对第一个任务的性能。

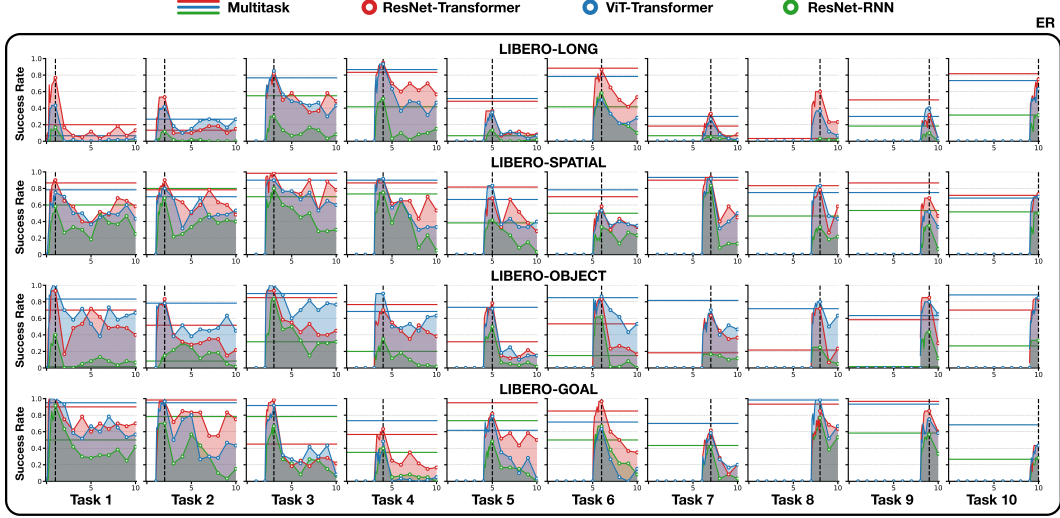


图 16: 使用 ER 算法对比不同架构。y 轴为成功率, x 轴显示智能体在学习过程中对给定任务套件中 10 个任务的性能。例如, 左上方的图显示智能体顺序学习 10 个任务时对第一个任务的性能。

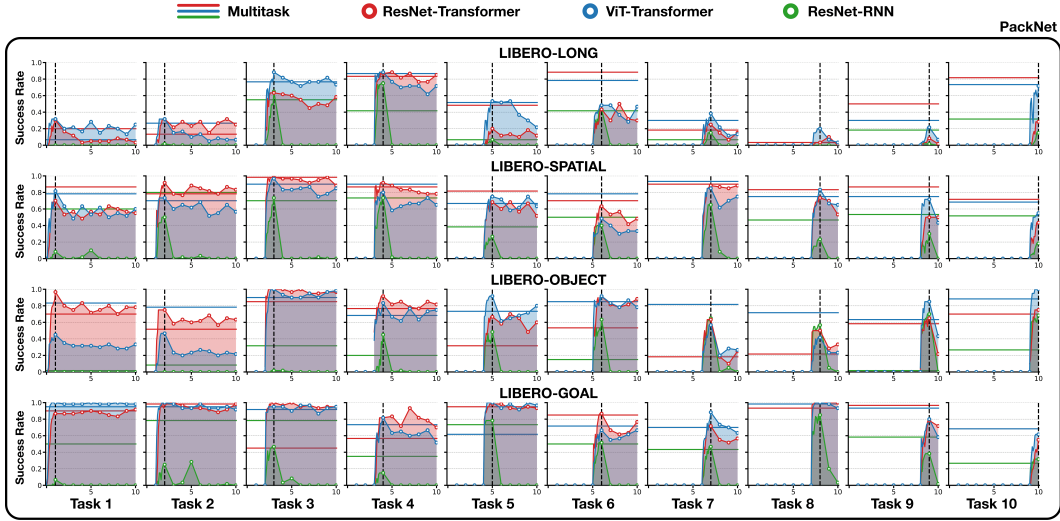


图 17: 使用 PACKNET 算法对比不同架构。y 轴为成功率, x 轴显示智能体在学习过程中对给定任务套件中 10 个任务的性能。例如, 左上方的图显示智能体顺序学习 10 个任务时对第一个任务的性能。

E.2 损失 v.s. 成功率

本节说明，以行为克隆损失衡量任务成功率可能产生误导。在监督学习任务（如图像分类）中，较低的损失通常表示更高的预测准确度。但这通常不适用于决策任务：误差会一步步累积，最终导致机器人执行失败 [55]。图 18、13 和 14 对比绘制了三种终身学习方法（ER、EWC 和 PACKNET）的训练损失与成功率；每种方法都在三种神经架构、四个任务套件上评估。

发现：我们观察到虽然有时 EWC 有**最低**的损失，但它没有取得良好的成功率。另一方面，ER 可能有最高的损失但表现优于 EWC。因此应以成功率而非行为克隆损失作为模型检查点优劣的评估指标。

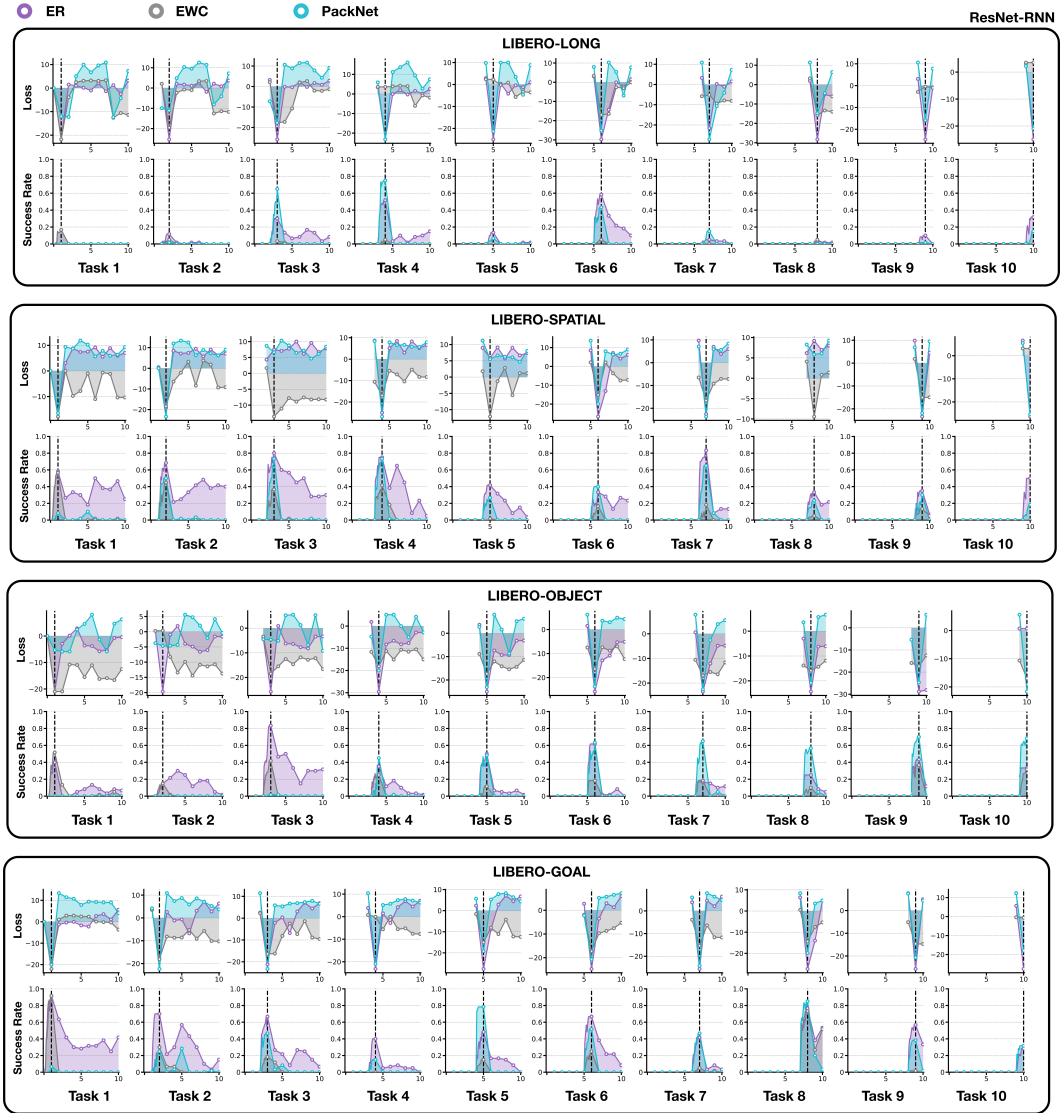


图 18: ER (紫色)、EWC (灰色) 和 PACKNET (蓝色) 在具有 RESNET-RNN 策略的四个任务套件上的损失和成功率。第一行(第二行)显示智能体在整个 LLDMM 过程中对任务 i 的损失(成功率)。

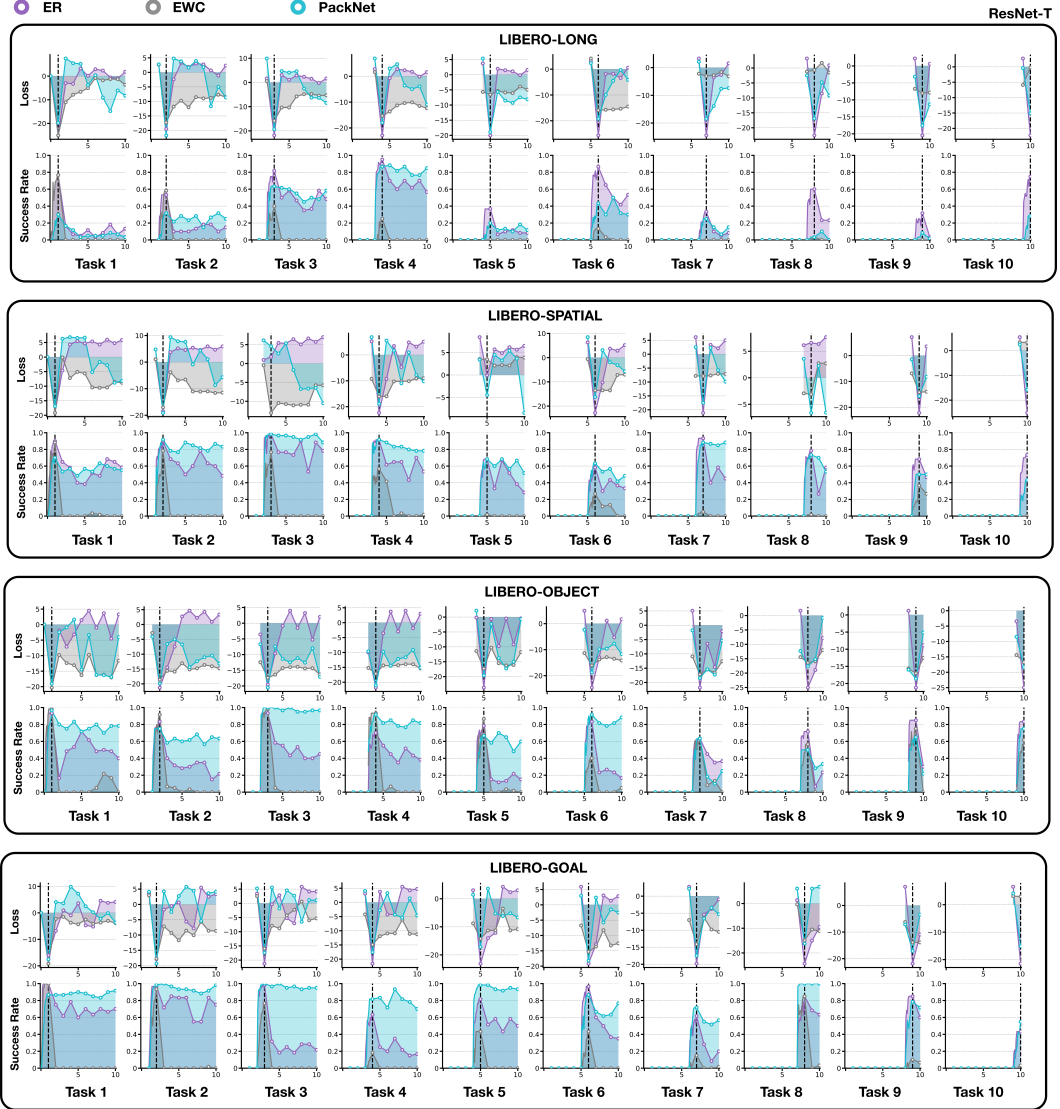


图 19: ER (紫色)、EWC (灰色) 和 PACKNET (蓝色) 在具有 RESNET-T 策略的四个任务套件上的损失和成功率。第一行 (第二行) 显示智能体在整个 LLDM 过程中对任务 i 的损失 (成功率)。

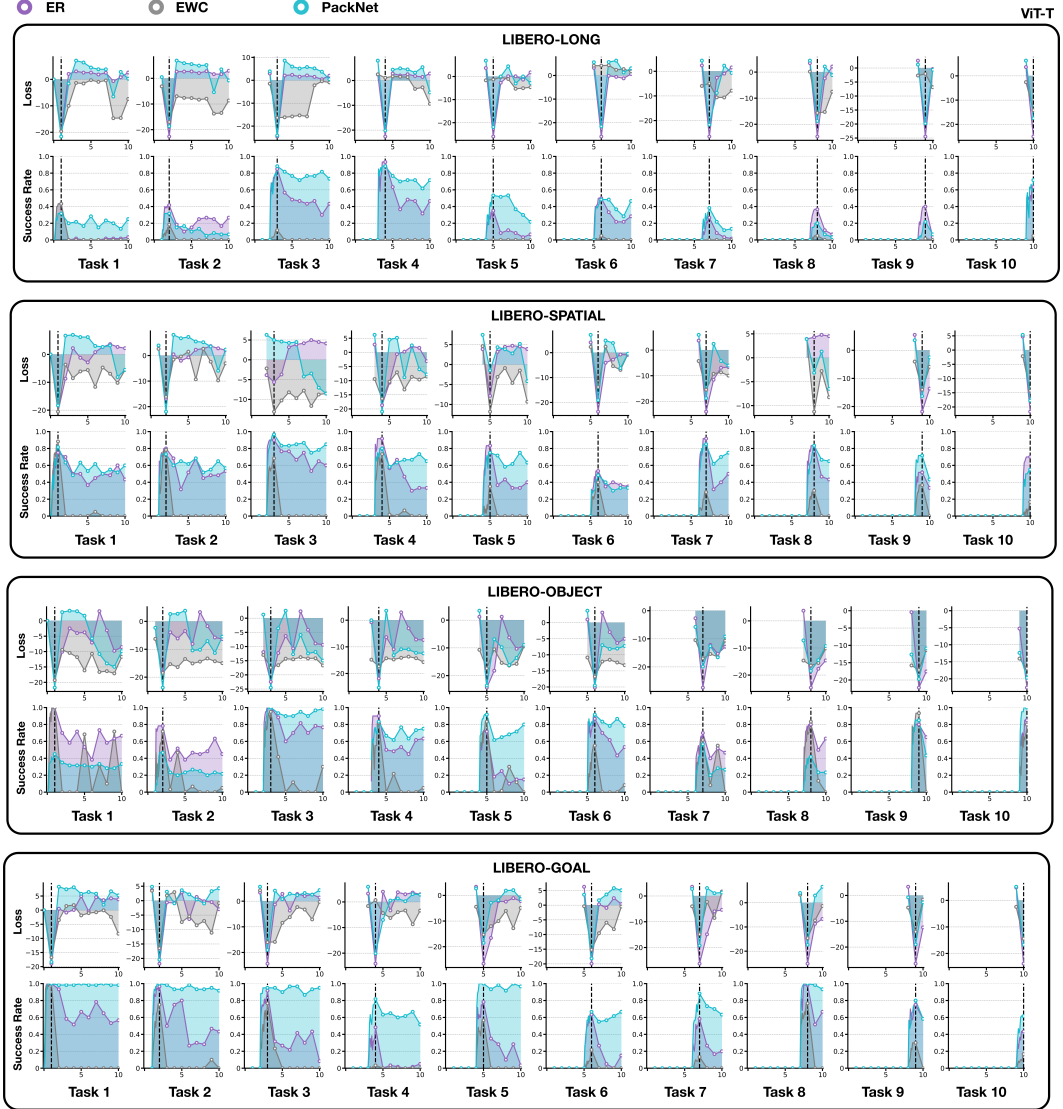


图 20: ER (紫色)、EWC (灰色) 和 PACKNET (蓝色) 在具有 ViT-T 策略的四个任务套件上的损失和成功率。第一行 (第二行) 显示智能体在整个 LLDM 过程中对任务 i 的损失 (成功率)。

E.3 多任务成功率

Env Name	RESNET-RNN	RESNET-T	ViT-T
KITCHEN SCENE10 close the top drawer of the cabinet	0.45	0.45	0.6
KITCHEN SCENE10 close the top drawer of the cabinet and put the black bowl on top of it	0.1	0.1	0.0
KITCHEN SCENE10 put the black bowl in the top drawer of the cabinet	0.0	0.25	0.05
KITCHEN SCENE10 put the butter at the back in the top drawer of the cabinet and close it	0.15	0.0	0.0
KITCHEN SCENE10 put the butter at the front in the top drawer of the cabinet and close it	0.2	0.0	0.0
KITCHEN SCENE10 put the chocolate pudding in the top drawer of the cabinet and close it	0.25	0.0	0.0
KITCHEN SCENE1 open the bottom drawer of the cabinet	0.05	0.0	0.15
KITCHEN SCENE1 open the top drawer of the cabinet	0.3	0.45	0.25
KITCHEN SCENE1 open the top drawer of the cabinet and put the bowl in it	0.05	0.0	0.0
KITCHEN SCENE1 put the black bowl on the plate	0.35	0.3	0.0
KITCHEN SCENE1 put the black bowl on top of the cabinet	0.2	0.7	0.15
KITCHEN SCENE2 open the top drawer of the cabinet	0.45	0.4	0.65
KITCHEN SCENE2 put the black bowl at the back on the plate	0.2	0.05	0.35
KITCHEN SCENE2 put the black bowl at the front on the plate	0.1	0.35	0.0
KITCHEN SCENE2 put the middle black bowl on the plate	0.35	0.1	0.0
KITCHEN SCENE2 put the middle black bowl on top of the cabinet	0.5	0.1	0.6
KITCHEN SCENE2 stack the black bowl at the front on the black bowl in the middle	0.25	0.05	0.25
KITCHEN SCENE2 stack the middle black bowl on the back black bowl	0.05	0.05	0.0
KITCHEN SCENE3 put the frying pan on the stove	0.4	0.3	0.35
KITCHEN SCENE3 put the moka pot on the stove	0.15	0.0	0.4
KITCHEN SCENE3 turn on the stove	0.6	0.7	1.0
KITCHEN SCENE3 turn on the stove and put the frying pan on it	0.15	0.0	0.35
KITCHEN SCENE4 close the bottom drawer of the cabinet	0.75	0.4	0.55
KITCHEN SCENE4 close the bottom drawer of the cabinet and open the top drawer	0.2	0.0	0.05
KITCHEN SCENE4 put the black bowl in the bottom drawer of the cabinet	0.25	0.3	0.25
KITCHEN SCENE4 put the black bowl on top of the cabinet	0.8	0.6	0.85
KITCHEN SCENE4 put the wine bottle in the bottom drawer of the cabinet	0.0	0.0	0.35
KITCHEN SCENE4 put the wine bottle on the wine rack	0.05	0.0	0.2
KITCHEN SCENE5 close the top drawer of the cabinet	0.05	0.8	0.9
KITCHEN SCENE5 put the black bowl in the top drawer of the cabinet	0.2	0.1	0.15
KITCHEN SCENE5 put the black bowl on the plate	0.05	0.05	0.1
KITCHEN SCENE5 put the black bowl on top of the cabinet	0.5	0.25	0.2
KITCHEN SCENE5 put the ketchup in the top drawer of the cabinet	0.0	0.0	0.05
KITCHEN SCENE6 close the microwave	0.1	0.1	0.2
KITCHEN SCENE6 put the yellow and white mug to the front of the white mug	0.2	0.05	0.1
KITCHEN SCENE7 open the microwave	0.7	0.1	0.45
KITCHEN SCENE7 put the white bowl on the plate	0.05	0.0	0.05
KITCHEN SCENE7 put the white bowl to the right of the plate	0.05	0.05	0.15
KITCHEN SCENE8 put the right moka pot on the stove	0.15	0.0	0.1
KITCHEN SCENE8 turn off the stove	0.2	0.25	0.75
KITCHEN SCENE9 put the frying pan on the cabinet shelf	0.45	0.15	0.05
KITCHEN SCENE9 put the frying pan on top of the cabinet	0.25	0.4	0.3
KITCHEN SCENE9 put the frying pan under the cabinet shelf	0.15	0.45	0.15
KITCHEN SCENE9 put the white bowl on top of the cabinet	0.1	0.1	0.15
KITCHEN SCENE9 turn on the stove	0.5	0.4	0.95
KITCHEN SCENE9 turn on the stove and put the frying pan on it	0.0	0.0	0.0

Env Name	RESNET-RNN	RESNET-T	ViT-T
LIVING ROOM SCENE1 pick up the alphabet soup and put it in the basket	0.0	0.0	0.0
LIVING ROOM SCENE1 pick up the cream cheese box and put it in the basket	0.0	0.0	0.0
LIVING ROOM SCENE1 pick up the ketchup and put it in the basket	0.0	0.0	0.0
LIVING ROOM SCENE1 pick up the tomato sauce and put it in the basket	0.0	0.0	0.0
LIVING ROOM SCENE2 pick up the alphabet soup and put it in the basket	0.0	0.0	0.0
LIVING ROOM SCENE2 pick up the butter and put it in the basket	0.0	0.0	0.0
LIVING ROOM SCENE2 pick up the milk and put it in the basket	0.0	0.05	0.0
LIVING ROOM SCENE2 pick up the orange juice and put it in the basket	0.0	0.0	0.0
LIVING ROOM SCENE2 pick up the tomato sauce and put it in the basket	0.0	0.05	0.0
LIVING ROOM SCENE3 pick up the alphabet soup and put it in the tray	0.0	0.05	0.0
LIVING ROOM SCENE3 pick up the butter and put it in the tray	0.0	0.3	0.0
LIVING ROOM SCENE3 pick up the cream cheese and put it in the tray	0.0	0.25	0.0
LIVING ROOM SCENE3 pick up the ketchup and put it in the tray	0.0	0.0	0.0
LIVING ROOM SCENE3 pick up the tomato sauce and put it in the tray	0.0	0.25	0.0
LIVING ROOM SCENE4 pick up the black bowl on the left and put it in the tray	0.0	0.4	0.2
LIVING ROOM SCENE4 pick up the chocolate pudding and put it in the tray	0.0	0.2	0.25
LIVING ROOM SCENE4 pick up the salad dressing and put it in the tray	0.0	0.0	0.1
LIVING ROOM SCENE4 stack the left bowl on the right bowl and place them in the tray	0.0	0.0	0.0
LIVING ROOM SCENE4 stack the right bowl on the left bowl and place them in the tray	0.0	0.0	0.05
LIVING ROOM SCENE5 put the red mug on the left plate	0.0	0.0	0.05
LIVING ROOM SCENE5 put the red mug on the right plate	0.15	0.0	0.0
LIVING ROOM SCENE5 put the white mug on the left plate	0.1	0.15	0.05
LIVING ROOM SCENE5 put the yellow and white mug on the right plate	0.35	0.05	0.05
LIVING ROOM SCENE6 put the chocolate pudding to the left of the plate	0.1	0.65	0.0
LIVING ROOM SCENE6 put the chocolate pudding to the right of the plate	0.05	0.55	0.0
LIVING ROOM SCENE6 put the red mug on the plate	0.0	0.2	0.0
LIVING ROOM SCENE6 put the white mug on the plate	0.0	0.2	0.0
STUDY SCENE1 pick up the book and place it in the front compartment of the caddy	0.0	0.0	0.05
STUDY SCENE1 pick up the book and place it in the left compartment of the caddy	0.2	0.05	0.0
STUDY SCENE1 pick up the book and place it in the right compartment of the caddy	0.0	0.1	0.0
STUDY SCENE1 pick up the yellow and white mug and place it to the right of the caddy	0.0	0.3	0.35
STUDY SCENE2 pick up the book and place it in the back compartment of the caddy	0.35	0.3	0.0
STUDY SCENE2 pick up the book and place it in the front compartment of the caddy	0.25	0.1	0.0
STUDY SCENE2 pick up the book and place it in the left compartment of the caddy	0.25	0.45	0.05
STUDY SCENE2 pick up the book and place it in the right compartment of the caddy	0.0	0.2	0.0
STUDY SCENE3 pick up the book and place it in the front compartment of the caddy	0.2	0.0	0.0
STUDY SCENE3 pick up the book and place it in the left compartment of the caddy	0.4	0.45	0.15
STUDY SCENE3 pick up the book and place it in the right compartment of the caddy	0.0	0.05	0.05
STUDY SCENE3 pick up the red mug and place it to the right of the caddy	0.0	0.2	0.05
STUDY SCENE3 pick up the white mug and place it to the right of the caddy	0.0	0.0	0.05
STUDY SCENE4 pick up the book in the middle and place it on the cabinet shelf	0.05	0.2	0.05
STUDY SCENE4 pick up the book on the left and place it on top of the shelf	0.2	0.1	0.15
STUDY SCENE4 pick up the book on the right and place it on the cabinet shelf	0.0	0.25	0.05
STUDY SCENE4 pick up the book on the right and place it under the cabinet shelf	0.15	0.1	0.05

E.4 注意力可视化

在终身学习过程中，可视化机器人完成任务时的行为与注意力图同样重要：它为比较不同算法与架构提供直觉和定性反馈。我们用 (author?) [23] 的方法可视化已学策略的注意力图，并在第 5.2 节的各项研究中对比，检查机器人是否关注了任务中该关注的区域。

基于扰动的注意力可视化： 我们使用基于扰动的方法 [23] 从智能体提取注意力图。给定输入图像 I ，该方法对像素位置 (i, j) 应用高斯滤波器以部分模糊图像，并生成扰动图像 $\Phi(I, i, j)$ 。将学习的策略表示为 π ，将空间模块（例如 resnet 或 ViT 编码器的最后潜在表示）对图像 I 的输入表示为 $\pi_u(I)$ 。然后我们将显著性分数定义为原始图像和模糊图像的潜在表示之间的欧几里得距离：

$$S_\pi(i, j) = \frac{1}{2} \left\| \pi_u(I) - \pi_u(\Phi(I, i, j)) \right\|^2. \quad (4)$$

直观地说， $S_\pi(i, j)$ 描述了从位置 (i, j) 周围区域移除信息对策略的影响程度。换句话说，较大的 $S_\pi(i, j)$ 表示像素 (i, j) 周围的信息对智能体的决策很重要。[23] 发现，不必逐像素计算分数，只在 $i \bmod 5$ 、 $j \bmod 5$ 的像素处计算显著性分数，就能在 Atari 游戏中以较低计算成本得到质量不错的显著性图。最终显著性图 P 归一化为 $P(i, j) = \frac{S_\pi(i, j)}{\sum_{i, j} S_\pi(i, j)}$ 。

可视化结果与分析见后续几页。

不同任务套件

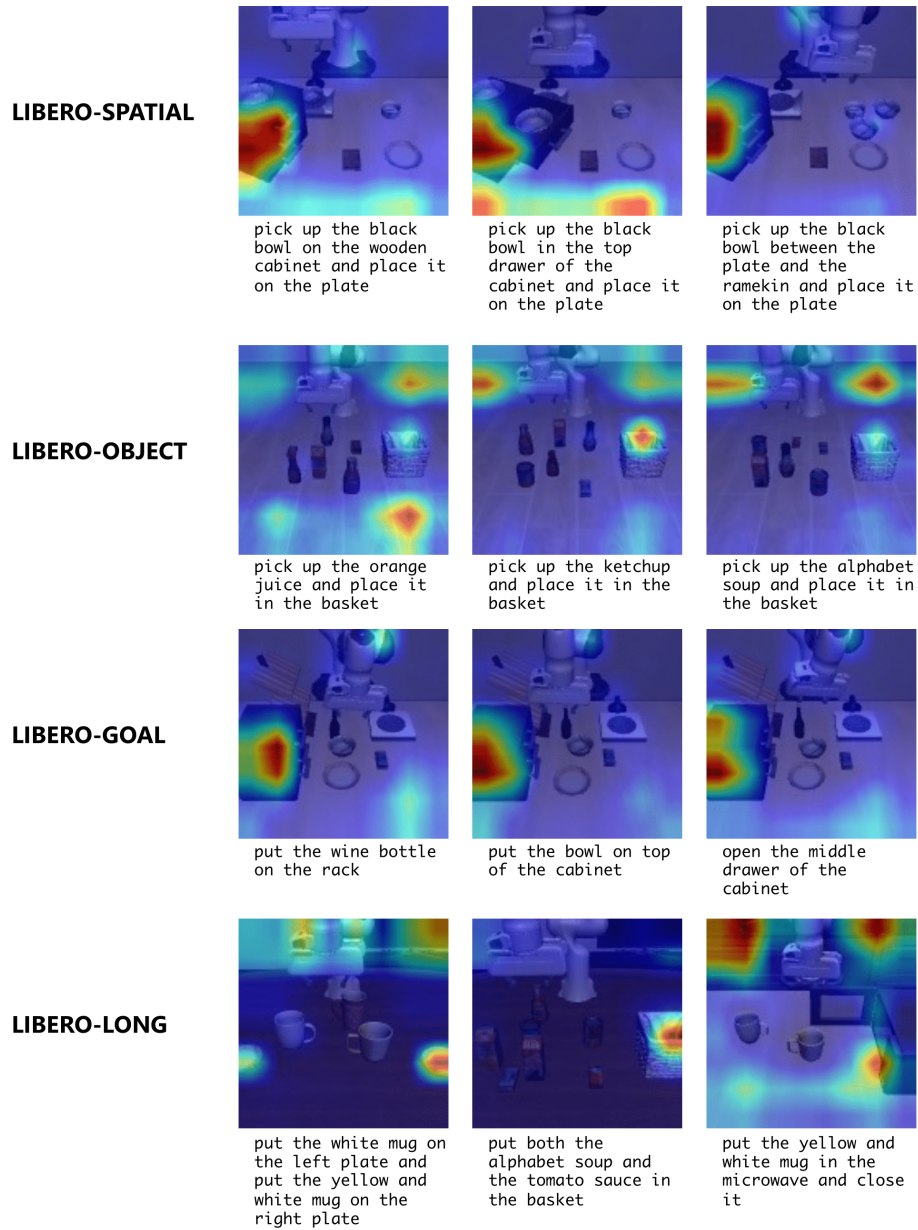


图 21: ER 和 RESNET-T 中不同任务套件的注意力图对比。每行对应一个任务套件。

发现： 图 21 显示了跨 4 个任务套件的 12 个任务的注意力可视化（例如，每个套件 3 个任务）。我们观察到：

1. 策略对机器人手臂和目标放置区域的关注度高于目标对象。
2. 有时策略会关注与任务无关的区域，例如桌子上的空白区域。

这些观察说明，策略做决策时利用感知数据的方式与人类很不一样。机器人策略常把与任务无关的特征虚假地关联到动作上——它因此过拟合于单个任务，难以跨任务泛化。

终身学习过程中的同一任务

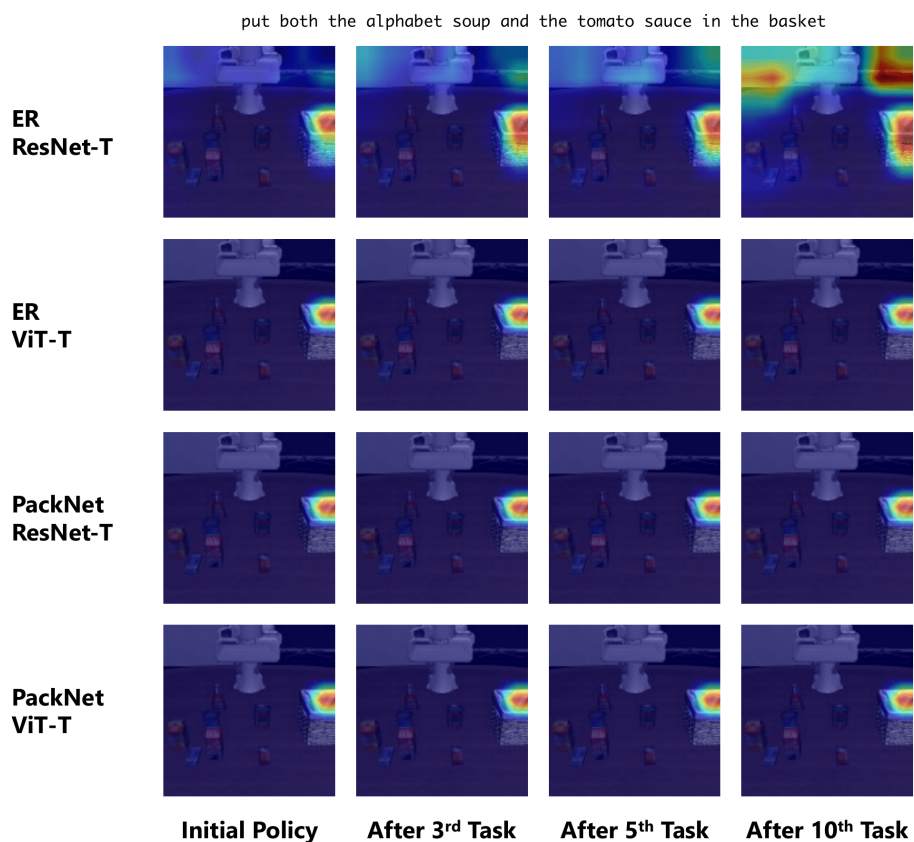


图 22: 在终身学习期间, LIBERO-LONG 中任务”将字母汤和番茄酱都放入篮子”的同一状态的注意力图。每行显示第一个任务上的注意力图如何随 LL 算法之一 (ER 和 PACKNET) 和神经架构之一 (RESNET-T 和 ViT-T) 变化。初始策略是在第一个任务上训练的策略。所有后续注意力图对应于在第三、第五和第十个任务上训练后的策略。

发现: Figure 22 shows attention visualizations from policies trained with ER and PACKNET using the architectures RESNET-T and ViT-T respectively. We observe that:

1. The ViT visual encoder’s attention is more consistent over time, while the ResNet encoder’s attention map gradually dilutes.
2. PackNet, as it splits the model capacity for different tasks, shows a more consistent attention map over the course of learning.

不同的终身学习算法

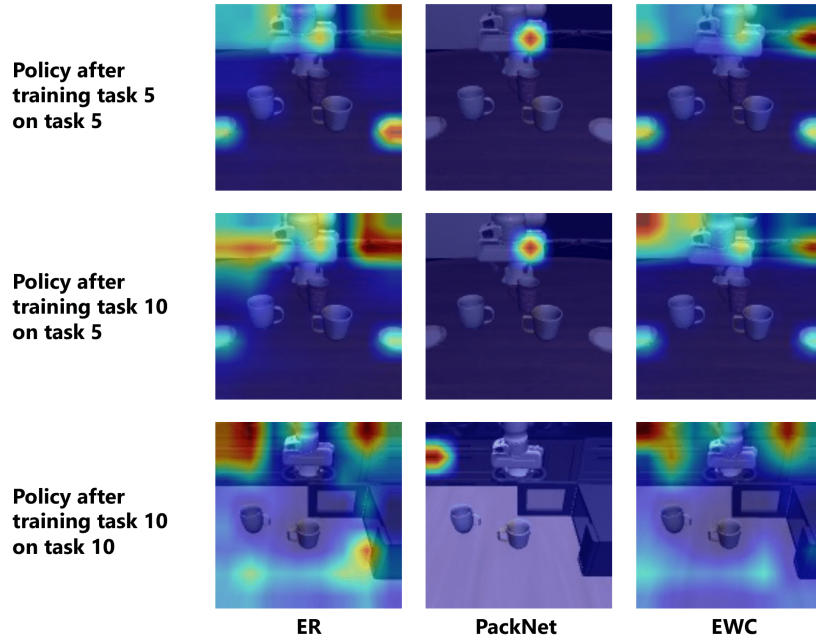


图 23: LIBERO-LONG 上具有 RESNET-T 的不同终身学习算法的注意力图对比。每行显示不同神经架构的任务的相同状态。”任务 5”指的是任务”将白色杯子放在左盘上，将黄白色杯子放在右盘上”。”任务 10”指的是任务”将黄白色杯子放入微波炉并关闭它”。第二行显示在任务 10 上训练并在任务 5 上评估的策略，显示后向迁移中的注意力图差异。

发现： Figure 23 shows the attention visualization of three lifelong learning algorithms on LIBERO-LONG with RESNET-T on two tasks (task 5 and task 10). The first and third rows show the attention of the policy on the same task it has just learned. While the second row shows the attention of the policy on the task it learned in the past. We observe that:

1. PACKNET shows more concentrated attention compared against ER and EWC (usually just a single mode).
2. ER shares similar attention map with EWC, but EWC performs much worse than ER. Therefore, attention can only assist the analysis but cannot be treated as a criterion for performance prediction.

不同的神经架构

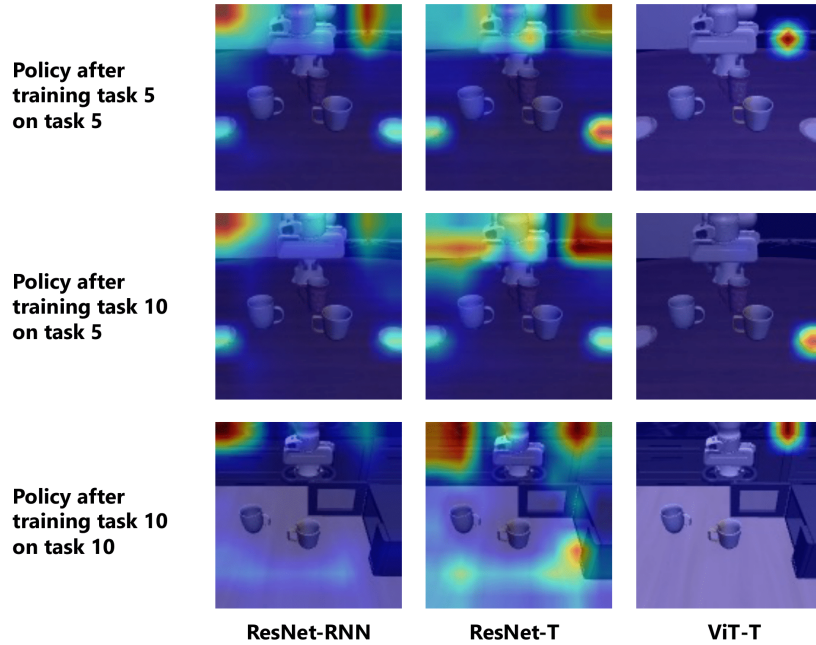


图 24: LIBERO-LONG 上具有 ER 的不同神经架构的注意力图对比。每行显示不同神经架构的任务的相同状态。”任务 5”指的是任务”将白色杯子放在左盘上，将黄白色杯子放在右盘上”。”任务 10”指的是任务”将黄白色杯子放入微波炉并关闭它”。第二行显示在任务 10 上训练并在任务 5 上评估的策略，显示后向迁移中的注意力图差异。

发现： Figure 24 shows attention map comparisons of the three neural architectures on LIBERO-LONG with ER on two tasks (task 5 and task 10). We observe that:

1. ViT has more concentrated attention than policies using ResNet.
2. When ResNet forgets, the attention is changing smoothly (more diluted). But for ViT, when it forgets, the attention can completely shift to a different location.
3. When ResNet is combined with LSTM or a temporal transformer, the attention hints at the "course of future trajectory". But we do not observe that when ViT is used as the encoder.

不同的任务顺序

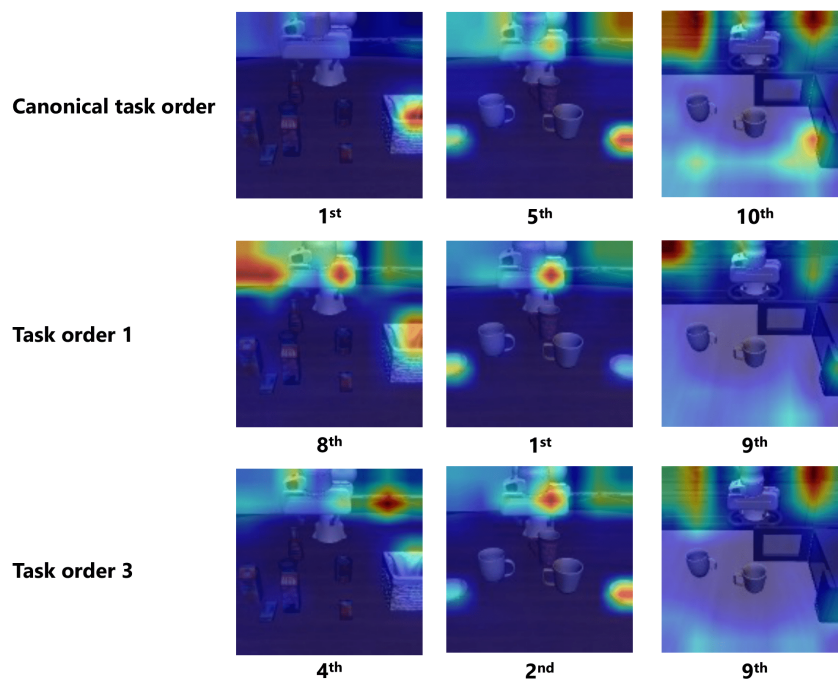


图 25: LIBERO-LONG 中三个选定任务上 ER 和 RESNET-T 的不同顺序的注意力图对比: 将字母汤和番茄酱都放入篮子、将白色杯子放在左盘上, 将黄白色杯子放在右盘上和 将黄白色杯子放入微波炉并关闭它。每行对应一个特定的任务顺序, 每个注意力图的标题表示该序列中任务的顺序。

发现：图 25 给出三种任务顺序下的注意力图对比, 展示的是用 ER 和 RESNET-T 在 LIBERO-LONG 上刚学完的两个任务。我们观察到:

1. 同一任务出现在任务流的不同位置, 注意力可视化确实不同, 符合预期。
2. 有一个趋势: 任务在序列中越靠后, 策略的注意力越分散。

有或没有预训练

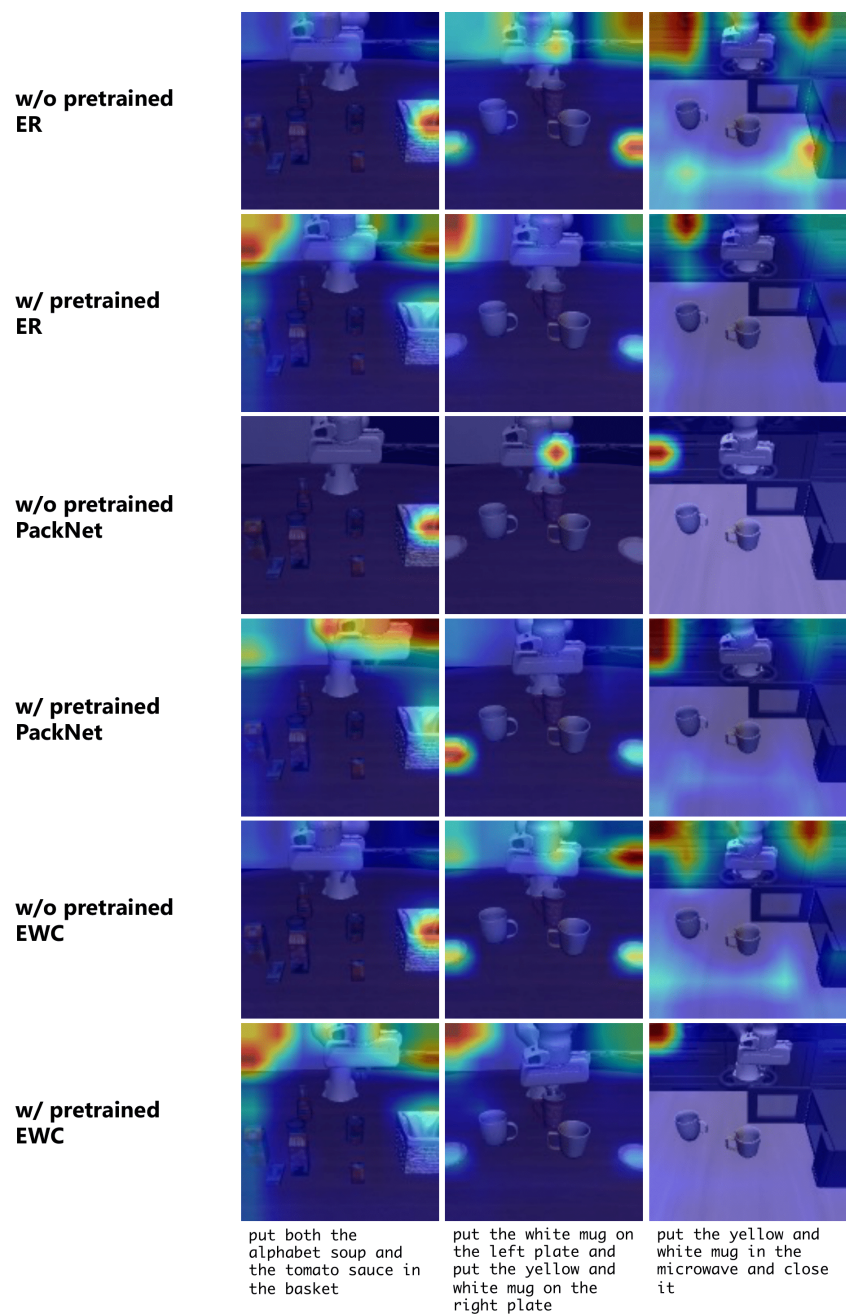


图 26: 使用 RESNET-T 和不同终身学习算法的无/有预训练模型的注意力图对比, 来自 LIBERO-LONG 的三个选定任务。

发现：图 26 给出 LIBERO-LONG 上 RESNET-T 搭配三种 LL 算法时，有无预训练两种模型的注意力图对比。我们观察到：

1. 经过预训练的策略，比未预训练的更容易被任务无关区域吸引。
2. 个别预训练策略对任务相关特征的注意力反而更好（第二行最后一个、第四行第二个），但性能仍然偏低。这再次说明：注意力图在语义上合理，并不意味着策略性能高。